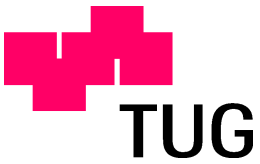
Implementierung eines SCORM-basierenden Zugangs zu Web-basierendem Unterricht

Magisterarbeit

Michael Mitter



26. September 2005

Implementierung eines SCORM-basierenden Zugangs zu Web-basierendem Unterricht

Magisterarbeit
an der
Technischen Universität Graz

vorgelegt von

Michael Mitter

Institut für
Informationssysteme und Computer Medien (IICM)
Technische Universität Graz
Inffeldgasse 16c, A-8010 Graz

September 2005

© 2005, Michael Mitter

Die Arbeit ist in deutscher Sprache verfasst.

Betreuer: Dipl.-Ing. Dr.techn. Denis Helic
Begutachter: Dr.Dr.h.c.mult.,O.Univ.-Prof. Hermann Maurer

Implementation of a SCORM-based approach to web-based Tutoring

Master's Thesis
at the
Graz University of Technology

submitted by

Michael Mitter

Institute for
Information Systems and Computer Media (IICM)
Graz University of Technology
Inffeldgasse 16c, A-8010 Graz

September 2005

© 2005, Michael Mitter

This work is written in German.

Supervisor: Dipl.-Ing. Dr.techn. Denis Helic
Assessor: Dr.Dr.h.c.mult.,O.Univ.-Prof. Hermann Maurer

Kurzfassung

E-Learning-Systeme werden zunehmend in Bildungseinrichtungen und Unternehmen zur Aus- und Weiterbildung eingesetzt. Wesentliche Vorteile gegenüber traditionellen Präsenz-Veranstaltungen liegen neben zeitlicher und räumlicher Flexibilität vor allem in den verschiedenen Betreuungsmöglichkeiten der Lernenden. Zusätzlich ist die Verwendung von Standards wie SCORM mittlerweile ein wichtiges Kriterium für den Erfolg und die Weiterentwicklung eines Systems.

Ziel dieser Magisterarbeit ist es, zunächst ein entsprechendes Web-basierendes Lern-Management-System zu implementieren, das Studenten die Möglichkeit bietet, SCORM-konforme Kurseinheiten auszuführen. Zudem sollen Konzepte gefunden werden, effiziente Tutoring-Möglichkeiten zu integrieren. Ansätze bestehen dabei in einer individuellen Anpassung von Lernpfaden in Verbindung mit entsprechenden Kommunikations- und Authoring-Möglichkeiten.

Einführend werden aktuelle Standardisierungsbestrebungen neben den veröffentlichten Standards und Spezifikationen vorgestellt. Danach erfolgt eine Evaluierung populärer kommerzieller und Open Source Lern-Management-Systeme bezüglich Funktionsumfang und Nutzung von Standards.

Weiterer wesentlicher Bestandteil der Arbeit ist eine Untersuchung der E-Learning-Spezifikation SCORM in ihrer aktuellsten Version. Initiiert von der Advanced Distributed Learning Initiative bildet diese zunehmend die Grundlage aktueller und zukünftiger E-Learning-Entwicklungen und -Szenarien. SCORM erfüllt dabei weitestgehend Anforderungen an E-Learning Standards wie Wiederverwendbarkeit und Interoperabilität.

Abstract

Educational institutions and even companies tend to choose e-learning systems for further education. The most obvious advantages of e-learning compared to traditional classroom teaching are temporal and local flexibility as well as the individual support provided for students. In addition, the application of standards, such as SCORM, has become an essential criterion of the success and further development of a system.

Basically, the main purpose of this master's theses is to implement a web-based learning management system which should enable students to execute SCORM compliant e-learning courses. Furthermore, concepts for providing efficient tutoring tools are to be presented. One possible approach consists of the individual adaptation of learning paths in combination with both a discussion board and an authoring tool.

At the beginning of this thesis current e-learning standardizing initiatives including published standard and specifications are being discussed. After that the two predominant types of learning management systems – commercial and open source – are evaluated according to the variety of tools and supported standards.

Another essential part of this work is evaluating the e-learning specification SCORM in its latest version. Launched by the Advanced Distributed Learning Initiative, SCORM is providing the basis of current and future developments in e-learning. Accordingly, this specification extensively satisfies the requirements of e-learning standards, such as reusability and interoperability.

Ich versichere hiermit, diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfsmittel bedient zu haben.

I hereby certify that the work presented in this thesis is my own and that work performed by others is appropriately cited.

Unterschrift des Autors:

Michael Mitter
Graz, Austria, September 2005

BibTeX

```
@mastersthesis{mitter:2005,
   author  = {Michael Mitter},
   title   = {{Implementierung eines SCORM-basierenden Zugangs zu Web-basierendem Unterricht}},
   school  = {Technische Universität Graz,
              Institut für Informationssysteme und Computer Medien (IICM)},
   address = {A-8010 Graz},
   year    = {2005},
   month   = {September},
   url     = {http://www.iicm.edu/thesis/mmitter.pdf}
}
```

© 2005, Michael Mitter

Der Autor übernimmt keine Haftung für Schäden, die aus der Verwendung dieses Dokuments und/oder eventuell auf einem Datenträger beigefügter Software entstehen.
Dieses Dokument enthält Verweise auf Websites, die von Dritten eingerichtet wurden. Der Autor hat keinerlei Kontrolle über die Websites und die dort angebotenen Informationen, Waren oder Dienstleistungen. Der Autor übernimmt daher keinerlei Verantwortung, aus welchem Rechtsgrund auch immer, für den Inhalt der Websites Dritter.
Die Lizenzbedingungen beigefügter Software entnehmen Sie bitte dem Installationsmedium und/oder dem Distributions-Archiv.

Danksagung

Die vorliegende Magisterarbeit wäre ohne die Unterstützung einiger Menschen nicht möglich gewesen. An dieser Stelle möchte ich mich bei all jenen bedanken, die mir beim Erstellen dieser Arbeit mit Rat und Tat zur Seite standen.

Der größte Dank gilt meinen Eltern, die mir das Studium ermöglichten und mir zu jeder Zeit jegliche Unterstützung bei der Erreichung meiner Ziele entgegenbrachten.

Weiters danke ich meinem Begutachter Dr.Dr.h.c.mult., O.Univ.-Prof. Hermann Maurer, Leiter des Instituts für Informationsverarbeitung und Computergestützte neue Medien und meinem Betreuer Dr. techn. Dipl.-Ing. Denis Helic.

Nicht zuletzt gilt mein Dank auch meinem Kollegen und Freund Thomas, der während dieser Arbeit und auch während des Studiums stets hilfreich und geduldig zur Seite stand.

Michael Mitter
Graz, Austria, September 2005

Credits

Kapitel 2 auf den Seiten 9–33 zum Thema „E-Learning: Tutoring Systeme“ wurde gemeinsam mit Thomas Knall verfasst und ist demnach ident mit dem gleichlautenden Kapitel der Arbeit „Automatische Adaptierung von SCORM-basierenden Lerninhalten“ ([Kna05]).

Inhaltsverzeichnis

Inhaltsverzeichnis i

Abbildungsverzeichnis vi

Tabellenverzeichnis viii

Quelltext-Verzeichnis ix

1 Einleitung 1

1.1 Motivation 1
1.2 Gliederung der Arbeit 6
1.2.1 Untersuchungsbereich 7
1.2.2 Gestaltungsbereich 7

I Untersuchungsbereich 8

2 E-Learning: Tutoring Systeme 9

2.1 Einleitung, Definition 9
2.2 Geschichtliche Entwicklung 10
2.3 Typen von E-Learning 12
2.4 Tutoring im speziellen 17
2.4.1 Tutoring Systeme 17
2.4.2 Intelligente Tutorielle Systeme 18
2.4.3 Persönliches Tutoring 20
2.4.4 Programmiertes versus persönliches Tutoring 21
2.5 Internationale E-Learning-Gremien und -Initiativen 23
2.5.1 Standardisierungsgremien und Standards 23
2.6 Fazit/Ausblick 32

3 SCORM 2004 **34**
3.1 Einführung . . . 34
3.1.1 ADL Initiative . . . 34
3.1.2 SCORM . . . 36
3.1.3 Geschichtliche Entwicklung . . . 38
3.2 Architektur . . . 39
3.2.1 Content Aggregation Model . . . 39
3.2.1.1 Content Model . . . 40
3.2.1.2 Content Packaging . . . 42
3.2.1.3 Meta-Data . . . 46
3.2.2 Run-Time Environment . . . 50
3.2.2.1 Launch Mechanism . . . 51
3.2.2.2 API . . . 53
3.2.2.3 Data Model . . . 56
3.3 Sequencing und Navigation . . . 57
3.3.1 Einführung . . . 58
3.3.1.1 Activity . . . 58
3.3.1.2 Activity Tree . . . 60
3.3.1.3 Attempts . . . 61
3.3.1.4 Learning Objectives . . . 61
3.3.2 Sequencing Definition Model . . . 61
3.3.2.1 Sequencing Control Modes . . . 63
3.3.2.2 Constrain Choice Controls . . . 65
3.3.2.3 Sequencing Rules . . . 65
3.3.2.4 Limit Conditions . . . 66
3.3.2.5 Rollup Rules . . . 66
3.3.2.6 Rollup Consideration Controls . . . 67
3.3.2.7 Selection Controls . . . 68
3.3.2.8 Randomization Controls . . . 68
3.3.2.9 Delivery Controls . . . 68
3.3.3 Tracking Model . . . 69
3.3.4 Overall Sequencing Process . . . 70
3.3.5 Navigation Model . . . 73
3.3.6 Sequencing and Navigation SCORM Version 1.2 . . . 73
3.3.7 Einschränkungen bzw. Unzulänglichkeiten . . . 75
3.4 Fazit und Ausblick . . . 76

4 Lern-Management-Systeme mit SCORM-Unterstützung 78
4.1 Einleitung . . . 78
4.1.1 Anforderungen . . . 79
4.2 Evaluierung . . . 81
4.2.1 SCORM Zertifizierung . . . 82
4.2.2 Kommerzielle Lernplattformen . . . 87
4.2.2.1 Blackboard . . . 87
4.2.2.2 CLIX . . . 90
4.2.2.3 WebCT . . . 92
4.2.3 Open Source Lernplattformen . . . 95
4.2.3.1 ILIAS . . . 97
4.2.3.2 ATutor . . . 100
4.2.3.3 Moodle . . . 102
4.2.4 Überblick der evaluierten LMS . . . 105
4.2.5 Standalone Applikationen . . . 107
4.3 Abschließende Betrachtungen . . . 109

II Gestaltungsbereich 111

5 Das Struts Framework 112
5.1 Das MVC-Paradigma . . . 112
5.1.1 Model 1 Architektur . . . 114
5.1.2 Model 2 Architektur . . . 114
5.2 Das Framework . . . 115
5.2.1 Model Komponenten . . . 115
5.2.1.1 Beans für die Geschäftslogik . . . 116
5.2.1.2 Beans für den Systemzustand . . . 116
5.2.1.3 ActionForm-Beans . . . 117
5.2.2 View Komponenten . . . 117
5.2.2.1 Struts-Tag-Libraries . . . 118
5.2.2.2 Internationalisierung . . . 118
5.2.3 Controller . . . 119
5.2.3.1 ActionServlet . . . 119
5.2.3.2 ActionMapping . . . 120
5.2.3.3 Action . . . 120
5.3 Modifikationen und Erweiterungen . . . 122
5.3.1 Struts Validator . . . 122

5.3.2 DynaActionForms . . . 123
5.3.3 Dynamic Transfer Object . . . 125
5.3.4 Struts Menu . . . 126
5.3.5 Tag Libraries . . . 127
5.3.5.1 JSTL . . . 127
5.3.5.2 Expression Language . . . 128
5.3.6 log4j . . . 129
5.3.7 Datenbank-Anbindung . . . 130
5.3.7.1 JDBC . . . 130
5.3.7.2 Database Connection-Pooling . . . 131
5.3.7.3 Prepared Statements . . . 133
5.4 Fazit . . . 133

6 Simple SCORM Player 135
6.1 Requirements . . . 135
6.2 Module . . . 137
6.2.1 Anmeldung . . . 137
6.2.2 Benutzerverwaltung . . . 138
6.2.3 Kursverwaltung . . . 140
6.2.4 Kurslektionen (Items) . . . 143
6.2.5 Diskussionsforum . . . 145
6.3 Individuelles Tutoring . . . 149
6.3.1 Problemstellung . . . 149
6.3.2 Konzept . . . 150
6.3.3 Möglicher Workflow . . . 151
6.4 Details zur Implementierung . . . 154
6.4.1 Struts Framework . . . 154
6.4.2 Paket-Hierarchie . . . 156
6.4.3 Datenbank-Struktur . . . 158
6.4.4 SCORM Integration . . . 160
6.4.4.1 Import von SCORM-Kursen . . . 160
6.4.4.2 Sequencing . . . 161
6.4.4.3 Einschränkungen . . . 162

7 Zusammenfassung und Ausblick 164

A Verwendete Software 167
A.1 Systemsoftware . . . 167

A.2 Pakete innerhalb der Laufzeitumgebung 168

B Dokumentation **170**

B.1 Einrichtung der Web-Applikation 170

B.1.1 Voraussetzungen 171

B.1.2 Installation 171

B.1.3 Konfiguration 172

B.1.4 Neukompilierung – build.xml `$CATALINA_HOME/webapps/SCORMplayer/` 177

Abkürzungsverzeichnis **185**

Glossar **196**

Literaturverzeichnis **197**

Index **210**

Abbildungsverzeichnis

2.1 Ideale Lernumgebung 14
2.2 Ideale Autorenumgebung 16
2.3 Prinzip eines tutoriellen Systems 18
2.4 Funktionsweise ITS 20
2.5 Kooperationsnetzwerk der Standardisierungsgremien 25
2.6 Unterstützte Standards in LMS 32
2.7 Bekanntheitsgrad von Standardisierungsinitiativen 33

3.1 Die ADL-Vision 36
3.2 Das SCORM-Bücherregal 38
3.3 SCORM Evolution 39
3.4 Geschichtliche Entwicklung von SCORM 40
3.5 Assets 41
3.6 Shareable Content Object 42
3.7 Komponenten des Content Aggregation Model 43
3.8 Content Hierarchie 44
3.9 Content Packaging 45
3.10 Manifest-Ressourcen 46
3.11 SCORM Laufzeitumgebung 50
3.12 API-Adapter Zustandsdiagramm 54
3.13 Schema eines Activity Trees mit Clustern 59
3.14 Ableitung des Activity Trees aus der Content Organization 60
3.15 Cluster Activity mit Choice 64
3.16 Cluster Activity mit Flow 64
3.17 Zusammenhang Tracking Model und RTE Data Model 70
3.18 Overall Sequencing Process 72

4.1 ADL SCORM 1.2 zertifizierte Produkte (Stand März 2005) 84

4.2 ADL SCORM 2004 zertifizierte Produkte (Stand März 2005) 86
4.3 Blackboard, SCORM-Kurs 89
4.4 Clix, Startseite 92
4.5 WebCT Vista, Interface für das Erstellen eines Kurses 95
4.6 ILIAS, Startseite („Persönlicher Schreibtisch“) 99
4.7 ATutor, Startseite 102
4.8 Moodle, Startseite 105
4.9 RELOAD Editor 2004 107

5.1 Model-View-Controller Prinzip 113
5.2 Model 2 Architektur des Struts Framework 115
5.3 Das Struts Validator Framework 123
5.4 Connection-Pooling mit JDBC 2.0 Extension API 132

6.1 Login-Ansicht 138
6.2 Benutzerverwaltung 139
6.3 Maske für das Erstellen eines neuen Benutzers 140
6.4 Ausführen eines Kurses 141
6.5 Kursübersicht 142
6.6 Editieren eines Kurses 142
6.7 Hinzufügen einer Kurslektion 144
6.8 Editieren einer Kurslektion 145
6.9 Erstellen einer Aktion 146
6.10 Diskussionsforum (MessageBoard) 148
6.11 Hinzufügen einer neuen Nachricht 148
6.12 MessageBoard, Thread-Ansicht 149
6.13 Anzeige ähnlicher Nachrichten 153
6.14 UML Diagramm - DTO, Java-Bean, Action 155
6.15 Prinzipielle Paket-Hierarchie des Simple SCORM Players 157
6.16 Vereinfachtes Datenbank-Schema 159
6.17 Festlegen von Sequencing-Attributen 162
6.18 Menüstruktur (Struts-Menu) 162
6.19 Navigationselemente 162

Tabellenverzeichnis

3.1 Metadaten-Anwendungsprofile im CAM 47
3.2 SCORM 2004 Navigations-Ereignisse 74

4.1 LMS und unterstützte Standards 106

5.1 JSTL Tag-Bibliotheken 127
5.2 log4j Prioritätslevel 129
5.3 log4j Appender 130

6.1 Pakete des Simple SCORM Players 156
6.2 MySQL-Tabellen des Simple SCORM Players 158

Quelltext-Verzeichnis

3.1 Beispiel einer Manifestdatei (`imsmanifest.xml`) 47
3.2 Beispiel für Metadaten 49
3.3 Beispiel für API-Funktionsaufrufe - Initialisierung und Beendigung der Kommunikation zwischen LMS (Learning-Management-System) und SCO (Shareable Content Object) 55
3.4 Beispiel für API-Aufruf eines SCOs mit der API-Funiktion `SetValue()`, dem Parameter `cmi.score.scaled` und dem Score `5` . . 57
3.5 Beispiel für Sequencing-Informationen in der Manifest-Datei 62
5.1 Beispiel für ein ActionMapping (`struts-config.xml`) 121
5.2 Beispiel für eine Action-Klasse (`MessageAction.java`) 122
5.3 Beispieleintrag in der Konfigurationsdatei des Validators (`validation.xml`) 124
5.4 Definintion einer DynaActionForm innerhalb der Struts-Konfigurationsdatei 125
5.5 Beispiel einer Menüdefinition (`menu-config.xml`) 126
5.6 Aufruf des Menüs innerhalb einer JSP-Datei 127
5.7 Beispiel für EL-Tags 128
5.8 Beispiel für Prepared Statement 133
B.1 Bekanntgabe der MySQL-Zugangsdaten über `struts-config.xml` . . 173
B.2 Konfiguration des SSP über `SCORMplayer.properties` 176
B.3 Konfiguration des Klassifikators über `Classifier4J.properties` . . 177

Kapitel 1

Einleitung

> Es ist des Lernens kein Ende.
>
> (Robert Schumann (1810-56), dt. Komponist)

1.1 Motivation

Das von Schumann getätigte Zitat im vorigen Jahrtausend hat nichts an seiner Gültigkeit verloren. Seit jeher ist der Mensch bestrebt, Wissen und Informationen zu sammeln und aufzunehmen. Heutige Gesellschaftsformen sind einem ständigen Wechsel, einer schnellen Evolution des Wissens unterworfen. Es wird gefordert, sich stetig weiterzubilden, vorhandenes Wissen wird immer schneller obsolet. Vielfach spricht man auch von einer Informationsgesellschaft. Längst befindet man sich in einem Prozess des lebenslangen Lernens, dies verdeutlicht, dass permanentes Lernen unverzichtbarer Bestandteil ist, um auch erfolgreich zu sein. Lernen hört somit nicht nach Schule, Ausbildung oder Studium auf, es ist das wesentliche Werkzeug zum Erlangen von Bildung und damit auch für die Gestaltung individueller Lebens- und Arbeitschancen.

Lernen

Der Begriff des Lernens hat sich im Laufe der Zeit in unserem Sprachgebrauch gefestigt und wird für vielfältige Handlungen benutzt, z.B. „Lernen in der Schule“ oder auch „Ein neues Land kennen lernen“. Es ist jedoch mehr als nur ein reines Abspeichern von Informationen. Im weitesten Sinn umfasst Lernen immer ein Handeln und Verhalten und wie dieses vor einem Lernprozess war bzw. wie es danach ist.

Ein Lernprozess kann dabei aus verschiedenen Stufen bestehen und ist abhängig von Vorkenntnissen, Erfahrung und Schwierigkeiten eines Lernenden. Am Beginn jedes Prozesses steht die Aufnahme von Wissen – es sollte einem möglich sein, Wörter und Fakten von sich wiederzugeben. Weitere Schritte sind, vorgegebene Information zu verstehen und anzuwenden. Schlussendlich sollte man in der Lage sein, eine Situtation zu analysieren, Schlussfolgerungen zu treffen und eine Urteilsabschätzung abzugeben.

Vielfach wird in diesem Zusammenhang auch der Begriff „Wissenserwerb“ verwendet. Unter Wissen versteht man im weitesten Sinne all das, was ein Mensch bei seinem Handeln heranzieht, also Entscheidungen zu Grunde legt. Im engeren Sinne handelt es sich um Erkenntnisse, die durch (eigene) Erfahrung oder durch vermitteltes Lernen erworben werden. Wissen kann sich also explizit in Dokumenten und implizit in den Köpfen der Menschen befinden. Information hingegen ist potenziell oder tatsächlich vorhandenes, nutzbares oder genutztes Wissen.

E-Learning

Hand in Hand mit dem technischen Fortschritt, beginnend mit der Entwicklung des Computers, hat sich auch das Lernen und damit auch die Wissensvermittlung angepasst und weiterentwickelt. Bereits vor mehreren Jahrzehnten bestand der Wunsch, Computer für die Aus- und Weiterbildung zu nutzen. Es fanden erste Versuche mit computergestütztem Lernen statt, scheiterten jedoch aufgrund unzureichender technischer Möglichkeiten. Erst mit entsprechender Hardware konnten vermehrt Methoden im Bildungsbereich erfolgreich umgesetzt werden. Mit der Entwicklung und Verbreitung von Netzwerken umfasste E-Learning ursprünglich alle Formen elektronisch gestützten Lernens, eine rasche Verbreitung des Internet bot eine Vielzahl an weiteren Möglichkeiten und Formen des E-Learnings. Bekannte Beispiele hierfür sind CBT (Computer Based Training) bzw. WBT (Web Based Training).

Grundsätzlich zeichnet sich E-Learning durch die Möglichkeit aus, orts- und zeitunabhängig zu lernen, Lernangebote bedarfsgerecht zu „portionieren“ und Kosten für Lernen zu reduzieren. Größter Vorteil des E-Learning ist die Flexibilität - ein Benutzer kann großteils selbst bestimmen von wo aus er an Kursen teilnehmen möchte.

Lern-Management-Systeme

Eine Nutzung und Bereitstellung von E-Learning-Bildungsangeboten setzt das Vorhandensein von so genannten Learning-Management-Systemen, auch Lernplattformen genannt, voraus. Prinzipiell hat sich trotz dem breiten Spektrum an Lern-Management-Systemen bzw. Online-Bildungsplattformen eine gewisse Basisfunktionalität herausgebildet, die viele Systeme aufweisen. Ein LMS (Learning-Management-System) stellt Instrumente für kooperatives Arbeiten sowie eine rollenbasierte Benutzerverwaltung zur Verfügung. Die verschiedenen Werkzeuge einer Lernplattform ermöglichen es, Lerninhalte anzusehen und auszuführen, Lernaktivitäten von Benutzern zu verfolgen und zu dokumentieren. Zentraler Bestandteil stellt die Unterstützung der Kommunikationen unter den verschiedenen Benutzerrollen dar.

Nach anfänglicher Euphorie stellte sich jedoch bald heraus, dass für einen erfolgreichen Einsatz von E-Learning-Anwendungen nicht unbedingt nur der Funktionsumfang sondern auch eine gewisse Kompatibilität ausschlaggebend ist. Aufgrund der stetig steigenden Anzahl an Lernplattformen, war der Markt an LMS bald sehr unübersichtlich. Wichtigstes Anliegen war, dass oftmals mit hohen Kosten erstellte Lerninhalte wiederverwendbar und auf verschiedenen Systemen nutzbar sein sollten. Somit war eine Entwicklung und Einführung von Spezifikationen und Standards im Bereich E-Learning unumgänglich.

Standards

Zwischenzeitlich hat sich eine größere Anzahl an Standardisierungsbestrebungen zu verschiedenen Bereichen des E-Learning entwickelt. Ziel der Gremien ist es, Standards zur Gewährleistung der pädagogischen Qualität der Lehre zu etablieren sowie diese kontinuierlich weiterzuentwickeln.

Hauptaufgabe eines Standards bzw. Spezifikation [1] ist es, Prozesse zu vereinfachen und zu vereinheitlichen, um so auch mit anderen Systemen zusammenwirken zu können – er stellt gewisse Mindesteigenschaften dar. Als grundlegende Eigenschaften gelten Wiederverwendbarkeit - Inhalte sollten in anderen Programmen und Kontexten verwendet werden können. Diese Inhalte sollen von verschiede-

[1] Sofern ein Standard noch nicht durch eine offizielle Standardisierungsinstitution bestätigt wurde, spricht man von einer Spezifikation.

nen Personen von unterschiedlichen Standorten aus abgerufen und benutzt werden können. Weitere Eigenschaften stellen eine kontinuierliche Weiterentwicklung, Interoperabilität, Erschwinglichkeit und Anpassungsfähigkeit dar.

SCORM

Diese geforderten Eigenschaften versucht seit einigen Jahren die Advanced Distributed Learning (ADL) Initiative des amerikanischen Verteidiungsministeriums in einem gemeinsamen Referenzmodell zu vereinen, dem Sharable Content Object Reference Model, kurz SCORM. Das Referenzmodell stützt sich dabei auf Einzelspezifikationen, die von anderen Standardisierungsgremien, wie IMS, AICC, DCMI, IEEE und ARIADNE erarbeitet wurden.

Die Aufgabe von ADL besteht nun darin, einen sicheren Zugang zu qualitativ hochwertiger Bildung und Training, die auf individuelle Bedürfnisse ausgerichtet sind, zu gewährleisten. Kosteneffizienz sowie Unabhängigkeit von Zeit und Ort sind weitere Forderungen der Initiative. Besonderes Augenmerk wird dabei auf Netzwerk- und Web-basierende Kurse gelegt.

In der aktuellsten Version SCORM 2004 liegen vier Dokumente[2] vor, worin zusammenhängende und aneinander angepasste technische Spezifikationen dargestellt werden und jede einen Teilaspekt des Gesamtmodells realisiert. Neben einer XML-basierten Spezifikation zur Beschreibung und Darstellung von Kursstrukturen und Lernmaterialen (*Content Aggregation Model*) beinhaltet SCORM auch eine Spezifikation für eine Laufzeitumgebung einschließlich eines definierten Befehlssatzes (API). Mit diesem wird ein Datenaustausch zwischen Lerninhalt und Lernumgebung ermöglicht (*Run-Time Environment*). Der dritte Teil beschreibt die Festlegung von Metadaten über Lerninhalte (*Meta-Data Model*). Als viertes Dokument hinzugekommen ist die Spezifikationen zur Steuerung der Abfolge von Lerneinheiten in Abhängigkeit vom Anwenderverhalten und bisherigen Lernergebnissen (*Sequencing and Navigation*).

Mit der vorliegenden, aktuellsten Version erfüllt SCORM noch nicht alle gewünschten Anforderungen im E-Learning-Bereich; dennoch wird das Referenzmodell durch die bereits erreichten und nutzbaren Ergebnisse und die ständige Weiterentwicklung vermehrt zur Grundlage zukünftiger E-Learning-Entwicklungen und

[2] In Zusammenhang mit SCORM spricht man auch von „Bücher“.

-Szenarien.

Individuelles Tutoring

Die vorliegende Arbeit beschäftigt sich mit der Problemstellung, wie eine Unterstützung von Studenten bei der Ausführung von Kursinhalten implementiert werden kann.

Zunächst ist es jedoch erforderlich, festzustellen, welche Möglichkeiten und Lösungswege hinsichtlich der Problemstellung existieren:

Lernpfade: Prinzipiell ist ein Lernpfad bereits durch den Entwickler des jeweiligen E-Learning-Kurses starr vorgegeben. Wünschenswert wäre hier, Lernpfade individuell in kürzester Zeit anpassen zu können, um somit auf studentische Problemstellungen reagieren zu können.

Authoring/Autorensysteme: Die Erstellung und multimediale Aufbereitung von Lerninhalten erfolgt mit speziellen Autorenwerkzeugen. Im Kontext eines LMS existieren zwei Möglichkeiten: Zum einen bieten Lernumgebungen als Modul integrierte Authoring-Tools an, zum anderen werden externe, eigenständige Applikationen von Fremdherstellern verwendet.
Hierbei bestehen aber teilweise große Unterschiede in Funktionalität und Anwendung. Einerseits wird ein Überangebot an Funktionen geboten, wodurch es mitunter recht aufwendig und zeitintensiv ist, Inhalte zu erstellen. Andererseits ist es oftmals nur möglich, Inhalte mit entsprechenden Programmierkenntnissen (Pseudocode) zu erstellen. Zum Teil verzichten LMS sogar gänzlich auf ein Authoring-Werkzeug und stellen nur eine Importmöglichkeit mit zuvor generierten Templates zur Verfügung.
Weiters ist ein Hinzufügen von Inhalten zu bereits existierenden Kursen häufig umständlich, weil diese in sich geschlossene Einheiten darstellen. Auch bereitet schon der Import von standardisierten Kursen vielfach Probleme.

Eine einfache und effiziente Möglichkeit, Inhalte zu einem bestehenden Kurs hinzuzufügen ist somit oftmals nicht gegeben.

Kommunikationswerkzeuge: Viele LMS stellen umfangreiche Kommunikations-Tools zur Verfügung. Die Kommunikation zwischen Studenten und Tutoren kann dabei synchron via Chat bzw. asynchron mittels Diskussionsforen, Blackboards oder E-Mail erfolgen. Vielfach ist diese Kommunikation jedoch

nicht zwingend an einen Kurs gebunden und nicht individuell auf einzelne Studenten anpassbar. Ein Nachteil besteht auch häufig in der umständlichen und aufwendigen Bedien- und Administrierbarkeit.

Tests: Verschiedene Formen von Tests bieten einem betreuenden Lehrer die Möglichkeit, festzustellen, wann eine Anpassung des Kursmaterials notwendig ist. Ein großer Nachteil besteht aber darin, dass Tests erst ausgewertet werden müssen und meist am Ende eines Kurses oder einer Lerneinheit ausgeführt werden können – ein direktes Reagieren auf studentische Fragestellungen ist somit nicht möglich.

Feedback: Eine weitere Möglichkeit einer Kommunikation zwischen Studenten und Tutoren bieten Feedback-Formulare. Diese werden üblicherweise jedoch auch erst am Ende eines Kurses eingesetzt. Eine nachfolgende Rückmeldung des Tutors kommt meistens zu spät und ist daher nur bedingt als unterstützendes Instrument verwendbar.

Betrachtet man die angeführten Möglichkeiten, erscheint es am effizientesten, Lernpfade und Authoring als Ausgangpunkte weiterer Überlegungen zu wählen.

Ziel dieser Arbeit ist es somit, zunächst ein Lern-Management-System zu implementieren, das Studenten die Möglichkeit bietet, individuelle SCORM-Kurse auszuführen und Fragen zu einzelnen Kurs-Elementen zu stellen bzw. zu diskutieren. Tutoren sollen dabei eine effiziente Möglichkeit vorfinden, auf diese Fragen zu reagieren und den Studenten zu unterstützen – einerseits mit individueller Anpassung von Lernpfaden andererseits mittels entsprechender Unterstützung innerhalb eines Diskussionsforums.

1.2 Gliederung der Arbeit

Die vorliegende Arbeit beschäftigt sich allgemein mit E-Learning und im speziellen mit der Spezifikation SCORM. Sie unterteilt sich in zwei Bereiche: Der Untersuchungsbereich gibt eine Einführung in die Thematik E-Learning, SCORM 2004 und schließt mit einer Evaluierung populärer Lern-Management-Systemen ab. Darauf aufbauend wird im Gestaltungsbereich ein Beispiel-LMS implementiert unter besonderer Berücksichtigung der SCORM-Spezifikation und allgemeine Informationen die Architektur und das verwendete Framework betreffend erläutert.

1.2.1 Untersuchungsbereich

Nach dem Einführungskapitel werden im zweiten Kapitel neben der geschichtlichen Entwicklung wesentliche Begriffe und Defintionen in Zusammenhang mit E-Learning, Tutoring und Lern-Management-Systemen erläutert. Weiters werden wichtige Standardisierungsgremien, Spezifikationen und Standards diskutiert. Darauf aufbauend wird in Kapitel drei die von der ADL Initiative veröffentlichte Spezifikation SCORM in ihrer aktuellsten Version vorgestellt und genauer betrachtet. Insbesondere steht das so genannte SCORM Sequencing and Navigation im Mittelpunkt. Zentraler Bestandteil des vierten Kapitels ist eine überblicksmäßige Evaluierung und Gegenüberstellung kommerziell-proprietärer und Open Source Lernplattformen.

1.2.2 Gestaltungsbereich

In den abschließenden Kapiteln fünf und sechs erfolgt die Beschreibung einer prototypischen Umsetzung eines Lern-Management-Systems für die Darstellung von SCORM-konformen Inhalten. Die Implementierung wird vor allem mit Hilfe des Java-basierten Framework Struts umgesetzt. Das System beinhaltet neben typischen Funktionen eines Learning-Management-Systems auch Ansätze zur automatischen Adaptierung von Lerninhalten (siehe [Kna05]). Einführend werden noch die Architektur des LMS und Grundlagen des verwendeten Frameworks und damit verbunden, weitere Werkzeuge vorgestellt und diskutiert.

Schließlich erfolgt in Kapitel sieben eine Zusammenfassung dieser Arbeit und ein Ausblick auf zukünftige Entwicklungen hinsichtlich E-Learning(-Systemen).

Literatur: [uni], [Jen04], [inf], [HA04], [LS04a], [Che04], [Häf03], [Bau02], [ano05], [Ger04], [Pri03]

Teil I

Untersuchungsbereich

Kapitel 2

E-Learning: Tutoring Systeme

> E-Learning is learning in the digital age where technology is used to improve the learning. This would be not just over the web but might also be in the classroom.
>
> (Elliott Masie)

Im folgenden Kapitel erfolgt eine kurze Begriffsdefinition von E-Learning und dessen geschichtlicher Background. Weiters werden grundlegende Zusammenhänge und elementare Begriffe hinsichtlich E-Learning erläutert. Mit Tutoring wird eine bekannte Form des E-Learning vorgestellt und verschiedene Arten kurz betrachtet bzw. gegenübergestellt. Abschließend folgt eine Auflistung und Beschreibung wichtiger Standardisierungsgreminen samt deren veröffentlichten Standards und Spezifikationen.

2.1 Einleitung, Definition

E-Learning ist ein schillernder Begriff, der sowohl in wissenschaftlicher Literatur als auch von Firmen und (Bildungs-)Einrichtungen in ganz unterschiedlicher Weise definiert wird. Wörtlich übersetzt bedeutet E-Learning „elektronisches Lernen“, dementsprechend war es ursprünglich ein Sammelbegriff für alle Lern- und Lehrplattformen, die elektronisch durch Informations- und Kommunikationstechnologien unterstützt wurden.

Zu Beginn seines Auftretens konzentrierte es sich stärker auf elektronisch unterstütztes Lernen mittels interaktivem TV, CD-ROM und Videobändern, dem so

genannten CBT; häufig findet man in Zusammenhang mit E-Learning neben CBT auch den Begriff Computer Based Instruction (CBI).

Im Zuge des Internet-Hype wurde mit dem Begriff E-Learning hautpsächlich die Form des netzwerkbasierenden Lernens assoziiert, d.h. es werden Systeme in LAN, WAN, Internet, neuerdings auch Wireless/Mobile Access eingesetzt. Bei dieser netzgestützten Form des Lernens, dem so genannten WBT besteht der große Vorteil in der Online-Verfügbarkeit, Verteilung von beliebigen Inhalten und der Möglichkeit einer Online-Kommunikation bzw. -interaktion zwischen Lehrenden und Lernenden oder den Lernenden untereinander über diese Inhalte.

Sowohl CBT als auch WBT beschreiben meist in sich geschlossene Systeme. Mit der Entwicklung des Next Generation Web rücken jedoch LMS in den Vordergrund, welche aus vormaligen WBTs Lernportale entstehen lassen und aus technischer Sicht eine neue Herausforderung darstellen.

2.2 Geschichtliche Entwicklung

Die Entwicklung des E-Learning begann entsprechend früh mit ersten Konzepten von Lernprogrammen und lässt sich aus zwei Blickwinkeln betrachten: Einerseits aus **lerntheoretischer** Sicht, andererseits aus **technologischer** Sicht. Zwangsläufig ergibt sich eine Verknüpfung dieser beiden Sichtweisen, da theoretische Konzepte immer nur im Rahmen der technologischen Möglichkeiten umgesetzt werden können.

Die Entwicklung ist grob in drei Phasen unterteilbar:

1. **Phase:** Beginnend mit den fünfziger Jahren bis Mitte der Siebziger war diese Phase vor allem durch den Behaviourismus gekennzeichnet, dessen prominentester Vertreter der Psychologe B.F. Skinner[1] war.
   Lernen wird hier als Reaktion auf bestimmte äußere Reize aufgefasst, der Lernende hat eine passive Rolle, interne Prozesse werden nicht betrachtet. Der Lehrer hingegen wird als Autorität angesehen, der die Reihenfolge der Lern-

[1] Burrhus Frederic Skinner (1904-1990) war der prominenteste Vertreter des Behaviourismus in den USA und prägte den Begriff „operante Konditionierung“. Weiters erfand er das so genannte programmierte Lernen und verfasste den weltweit beachteten utopischen Roman „Futurum Zwei“.

inhalte starr und instruktiv vorgibt.

Technologisch gesehen war die Entwicklung des Transistors und damit verbunden die ersten Rechenmaschinen herausragend, wobei diese noch nicht in der Lage waren, komplexe Lernprogramme umzusetzen.

2. **Phase:** Diese dauerte bis Anfang der achtziger Jahre und war vorwiegend durch den Kognitivismus geprägt; im Unterschied zum Behaviourismus wird Lernen zu einer aktiven und selbständigen Verarbeitung von äußeren Reizen und kann als Wechselwirkung zwischen einem externen Angebot und der internen Struktur angesehen werden.

   In dieser Periode wurden auch die ersten Mikroprozessoren entwickelt, wodurch es zur Produktion der IBM-PCs mit dem Betriebssystem MS-DOS kam. Durch den wirtschaftlichen Erfolg und den stetig steigenden technischen Möglichkeiten konnten auch vermehrt Methoden im Bildungsbereich umgesetzt werden.

3. **Phase:** Die bis heute andauernde Phase war anfänglich geprägt von vermehrter Forschung in Bereichen der KI (Künstliche Intelligenz) und Entwicklung von ITS (Intelligentes Tutoren-System).
   Nach Eintreten einer gewissen Stagnation konzentrierte man sich mehr auf eine Verbesserung der bestehenden Konzepte durch vermehrte Benutzerfreundlichkeit und Adaptivität der vorhandenen Software.
   Aufgrund zunehmender Weiterentwicklung der technologischen Möglichkeiten gewann die multimediale Aufbereitung von Lerninhalten sowie das Lernen in vernetzten Umgebungen an Bedeutung. Auch eröffnete die rasche Verbreitung des Internets ab Mitte der neunziger Jahre eine Vielzahl an weiteren Möglichkeiten.

   Betrachtet man die lerntheoretische Ebene, so trat zusätzlich zum Kognitivismus der Konstruktivismus in Erscheinung, der ebenfalls interne Verstehensprozesse in den Mittelpunkt stellt. Eine Wechselwirkung zwischen internen und äußeren Prozessen besteht jedoch nicht. Der Wissenserwerb baut vorwiegend auf Vorwissen des Lehrenden in Verbindung mit sozialen Komponenten auf und kann als individueller Konstruktionsprozess gesehen werden.

   Die Umsetzung in Lernsysteme ist hier von komplexer Natur, jedoch aufgrund moderner, multimediafähiger Rechner und Verbreitung des Internets technisch relativ einfach realisierbar.

Gegenwärtig hat E-Learning bereits einen recht hohen Verbreitungsgrad - vor allem Firmen investieren viel in elektronische Weiterbildung und innerbetriebliche Schulungen in Form von CBTs. Neuere, technisch anspruchsvollere Anwendungsformen sind jedoch noch in geringem Maße vertreten, wobei WBTs aufgrund des Internets (hochwertig gestreamte Medien, Breitbandzugänge und fortschrittliches Webdesign) im Vormarsch sind.

2.3 Typen von E-Learning

Für die Beschreibung und Unterscheidung der Typen von E-Learning(-Szenarien) haben sich in der Fachwissenschaft keine verbindlichen begrifflichen Konventionen durchgesetzt, daher gibt es auch keine einheitliche Auflistung.

Grundsätzlich sind folgende Fachtypologien(-Szenarien) vermehrt anzutreffen:

Synchrones E-Learning: Die Wissensaufnahme und -vermittlung findet gleichzeitig statt, d.h. der Lernende und der Lehrende (oder Tutor) bzw. die Lernenden treffen sich virtuell zu einem festgelegten Zeitpunkt. Die Kommunikation erfolgt one-way als Broadcast oder interaktiv als Konferenz (Videokonferenz, Chat, Classroom- oder Application Sharing).

Asynchrones E-Learning: Basiert auf der Grundlage zeitunabhängiger Kommunikation- und Kooperations-Anwendungen. Die Teilnehmer (Lernende) und der Tutor können hier flexibel entscheiden, wann sie auf Anfragen reagieren und tauschen sich meist per E-Mail aus.

Telelearning (Teleteaching, Teletutoring): Umfasst sämtliche Lern- bzw. Lehrprozesse, welche mittels Informations- und Kommunikationstechnik an räumlich getrennten Lehr- und Lernorten durchgeführt werden. Im Gegensatz zum typischen Fernlernen ist eine kontinuierliche Kommunikation zwischen Lehrer und Lernenden bzw. den Lernenden untereinander zentraler Bestandteil des Telelearnings. Wesentliche Merkmale sind eine individuelle Auseinandersetzung mit Lerninhalten, multimediale Darstellung dieser Inhalte und Nutzung moderner Kommunikationstechniken zum sozialen Austausch über das Gelernte.

E-Learning mit Lernprogrammen und (zusammengestellten) Lernmateralien:

Wie einleitend bereits erwähnt, versteht man unter Computer Based Training aktives und selbständiges Lernen mittels einer Lernsoftware, die lokal (offline) auf einem Datenträger (meistens CD-ROM oder DVD) gespeichert vorliegt. Die lernende Person ist zeitlich und räumlich unabhängig, und hat keinen direkten Kontakt zu Lehrenden bzw. weiteren Lernenden.

Bei Web Based Training erfolgt der Zugriff auf die Lernsoftware via Internet, woraus sich wesentliche Vorteile ergeben wie: Ein direkter Kontakt zu Lehrenden und Mitlernenden, eine ständige Aktualisierung der Inhalte und eine weltweite, ortsunabhängige Verfügbarkeit.
Eine Einschränkung besteht jedoch bei Verwendung von Multimedia-Inhalten – nicht alle sind problemlos mit Hilfe des WWW darstellbar bzw. von der Bandbreite des Zugangs abhängig. Die gängigsten Synonyme für WBT sind WBL (Web Based Learning), Web Supported Teaching und WSL (Web Supported Learning).

Computer Supported Cooperative Learning (CSCL): Vorrangiges Ziel von CSCL ist es, die Zusammenarbeit im Team auch bei räumlicher Trennung zu unterstützen und mögliche Probleme zu vermeiden, die etwa bei physischen Treffen auftreten könnten. Beispiele für Problemsituationen sind: Dominanz bestimmter Teilnehmer in Gesprächen, kulturelle Voreingenommenheit, Gruppenzwang, zeitliche und finanzielle Schwierigkeiten.

Mittlerweile hat dieser Trend auch in der Lehre Einzug gehalten. Studien belegen nämlich, dass gemeinsame Lernaktivitäten am effizientesten sind. Durch Gruppenarbeit steht die Anwendung des Wissens im Vordergrund, wodurch es implizit zunächst erlangt werden muss.

Daraus resultierend hat sich das Computer Supported Cooperative Work (CSCW) gebildet. Hier ist eine Kommunikation sowohl unter den Lernenden als auch mit den Betreuern für das Verständnis des Stoffes wichtig. Die Kommunikation verläuft idealerweise synchron (Chats) bzw. auch asynchron (E-Mail, Diskussionsforen). Als hilfreich erweisen sich auch ein gemeinsamer Bereich zum Austausch von Daten und die Einrichtung von privaten Bereichen.

Eine alle Bildungsbereiche und Bedürfnisse abdeckende Typologie der didaktischen Grundmuster (Behaviourismus, Kognitivismus, Konstruktivismus) des E-Learning gibt es bislang nicht, auch führt eine einzelne Anwendung wahrschein-

lich nie zu einer Ideallösung.
Durch Entwicklung des Next Generation Web bieten so genannte LMS (Learning-Management-System), die aus vormaligen WBTs hervorgegangen sind, zunehmend die Möglichkeit, mehrere Arten unter einer einzigen Oberfläche zu integrieren.

Im Folgenden werden kurz Definitionen und Charakteristika von Learning-Management-Systemen erläutert:

Lernumgebung, Lernportal: Lernumgebungen vereinen mehrere Service-Dienste und sind nicht nur ein Portal für Lernende sondern auch für Administratoren (Lehrer und Tutoren inkludierend) und Autoren. In Abb. 2.1 wird das Aussehen einer Lernumgebung schematisch dargestellt und wie diese sich dem einzelnen Nutzer präsentiert:

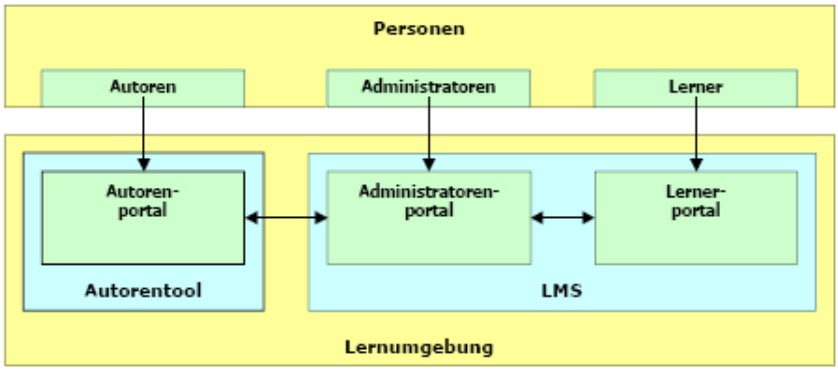


Abb. 2.1: Ideale Lernumgebung [Kai01]

Eine lernende Person findet ein auf sich abgestimmtes Portal vor. Es ermöglicht, Lehrmaterialien zu studieren, Fragen an einen Tutor zu stellen, mit anderen Leuten zusammen ein Dokument zu bearbeiten oder Testaufgaben zu absolvieren.
Der Tutor hat wiederum ein eigenes Portal mit entsprechenden Administrationsmöglichkeiten. Beispielsweise können Kurse zusammengestellt oder betreut und Lernfortschritte abgefragt werden.
Ein Autor benötigt ein eigenes, getrenntes Portal, worin er Inhalte generieren kann.

Lernplattform: Eine Lernplattform ist im Prinzip Teil einer Lernumgebung und ermöglicht via Internet/Intranet mit Hilfe einer entsprechenden Oberfläche, bestimmte administrative Funktionen durchzuführen.
Betrachtet man eine Lernplattform in Objekten, so lässt sich diese in drei Gruppen einteilen:

- Lehrinhalte
- Veranstaltungen
- Personen

Erstere werden durch Metadaten beschrieben und durch CMS (Content Management System) verwaltet, Veranstaltungen verfügen über Informationen einen Kurs betreffend und Personen werden mittels einer rollenbasierenden Teilnehmer-Datenbank verwaltet.
Weiters gibt es noch Prozesse, in welchen jene drei genannten Gruppen involviert sind. Prozesse sind beispielsweise: Administration, Personalisierung, Zeitsteuerung, Betreuungssystem oder Lernstatus.

Lehrinhalte, Autorensysteme: Eine weitere Funktionalität einer Lernplattform ist die Erstellung von Lehrinhalten. Für einen Autor ist es möglich, auf einfache Weise ohne besondere Programmierkenntnisse, Kursinhalte zu erstellen. Eine solche integrierte Lösung bringt aber auch einen großen Nachteil - fehlenden Flexibilität - mit sich.

Autorensysteme umgehen dies, indem diese vom restlichen System entkoppelt sind. Man unterscheidet zwischen seiten-, zeitachsen-, objekt- und struktogrammorientierten Systemen. Der große Unterschied liegt hier in der Programmierbarkeit, vorausgesetzt dem Autor ist die Programmiersprache des jeweiligen Systems bekannt.

Solche Systeme haben aber auch Nachteile:

- teilweise hoher Wartungsaufwand
- keine angemessene Visualisierung
- keine Unterstützung eigener didaktischer Konzepte
- hoher Aufwand für eine Zielgruppenanpassung
- kein Management verteilter Anwendungen

Autorensysteme werden daher nur im semiprofessionellen Bereich, beispielsweise an Universitäten, eingesetzt. Für professionelle Anwendungen bleibt

nur eine Arbeitsteilung als Lösungsweg: Ein didaktisches Konzept wird von einem Autor entwickelt, ein Programmierer setzt dieses um.

In Abb. 2.2 wird die prinzipielle Funktionsweise einer Autorenumgebung skizziert: Ein Autor stellt Konzepte und Daten in eine Datenbank, Programmierer und Grafiker können darauf zugreifen und entwickeln Templates für Lernressourcen und Lernobjekte. Mittels eines CMS hat ein Autor Zugriff auf Ressourcen und Objekte und kann diese beliebig variieren und zu einem Kurs zusammenstellen. Fertige Kurse werden an das LMS übermittelt. Zur Wahrung der Standardkonformität muss es dem CMS möglich sein, Ressourcen oder Kurs-Pakete nach Standardrichtlinien exportieren zu können.

LMS erweitert um Autorenwerkzeuge werden auch ILMS (Integrated Learning Management System) genannt.

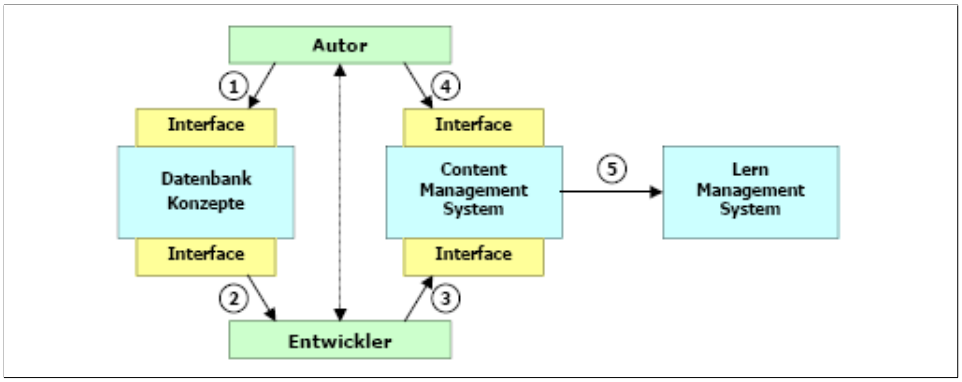


Abb. 2.2: Ideale Autorenumgebung [Kai01]

Learning Content Management System (LCMS): Ein LCMS kombiniert im Idealfall die Funktionalität eines LMS (Learning-Management-System) mit den Funktionen zur Content-Erstellung und zur Content-Personalisierung eines CMS (Content Management System). Es erlaubt das Erstellen von Lerninhalten und Lernobjekten durch mehrere Autoren. Weiters verfügt ein LCMS über eine Benutzerverwaltung, womit verschiedenen Personen(-gruppen) individuelle Rechte zugewiesen werden können. Ein weiterer wichtiger Aspekt ist die Unterstützung von wiederverwendbaren Lernobjekten, wodurch Redundanz und widersprüchliche Informationen weitgehend vermieden werden können.

Blended Learning bezeichnet eine Lehrmethode, bei der vor allem die Vorteile von Präsenzveranstaltungen und E-Learning systematisch eingesetzt bzw. verschmelzt werden. Das auch B-Learning genannte Konzept verbindet die Effektivität und Flexibilität von elektronischen Lernformen mit den sozialen Aspekten des gemeinsamen Lernens. Grundsätzlich ist somit auch ein LMS eine Blended-Learning-Anwendung, sofern CSCL-Komponenten mit Web-basierten Kursen verbunden werden. Hintergrund ist die Tatsache, dass das Lernen im Klassenzimmer immer noch von vielen als die angenehmste Lernmethode empfunden wird. Daher ist der zentrale Aspekt von Blended Learning eine Vor- bzw. Nachbereitung von Präsenzveranstaltungen; vor allem die Nachbereitung sichert einen gewissen Lerntransfer, welchen klassische Präsenzveranstaltungen nicht leisten können.

Zusammenfassend unterscheiden sich die verschiedenen Formen des E-Learnings insbesondere hinsichtlich folgender Aspekte:

- Grad der (elektronisch gestützten) Interaktivität
- Strukturierung der Inhalte beispielsweise nach fachlichen oder didaktischen Kriterien
- Grad der personellen Betreuung der Teilnehmerinnen und Teilnehmer
- Art des angestrebten Lernprozesses im Sinne von Arrangements „fertiger Inhalte“
- Einbeziehung traditioneller Formen des Lernens

2.4 Tutoring im speziellen

2.4.1 Tutoring Systeme

Tutorielle Lernumgebungen dienen hauptsächlich der Vermittlung von neuem Wissen und übernehmen daher typische Aufgaben, die sonst einem Lehrer bzw. Tutor vorbehalten sind. In diesem Zusammenhang fallen auch die bereits erwähnten Begriffe E-Tutoring, Tele-Tutoring bzw. WBT.
Tutorielle Lernprogramme (siehe Abb. 2.3 auf der nächsten Seite) übernehmen die

Lernfunktion, bieten systematisch Informationen, stellen Aufgaben, analysieren Antworten und sind in der Lage, gezielte Rückmeldungen an den Lernenden zu geben.

Weiters merkt sich das Programm den bisherigen Lernpfad eines Benutzers und gibt entsprechende Hinweise, welche Teile wiederholt oder welche zusätzlich konsultiert werden sollen. Ähnlich wie bei einem menschlichen Tutor findet also eine gewisse Lernsteuerung durch das Programm statt; umgekehrt hat auch der Benutzer mehr Eingriffsmöglichkeiten und ist nicht gezwungen, die Programmteile sequentiell abzuarbeiten. Der Ablauf ist jedoch nur durch die Antworten des Lernenden beeinflussbar.
Die größte Schwierigkeit bei der Kurserstellung liegt darin, dass der Kursablauf gänzlich im voraus geplant werden muss. Daher muss auf etwaige Verständnisprobleme seitens des Benutzers bereits bei der Kursplanung eingegangen werden.

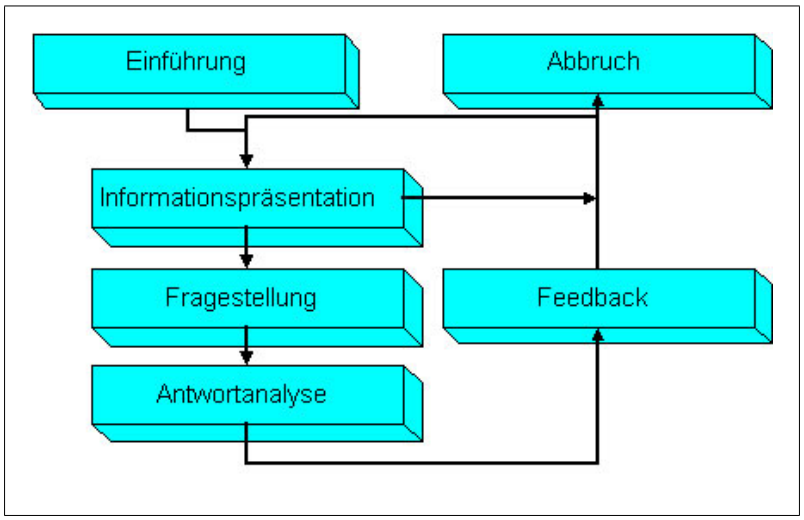


Abb. 2.3: Prinzip eines tutoriellen Systems [Blu98]

2.4.2 Intelligente Tutorielle Systeme

Im Kontext mit E-Tutoring wird des öfteren auch der Begriff Intelligentes Tutoren-System bzw. „Programmiertes Tutoring“ genannt. Intelligente Tutorielle Systeme

(ITS) (siehe Abb. 2.4 auf der folgenden Seite) versuchen vom Benutzer ein Modell abzuleiten und daraus eine individuelle Lehrstrategie für diesen zu entwickeln.

Es werden die Lösungsvorschläge der Lernenden analysiert und ein Modell über ihren Wissensstand und ihre Fähigkeiten erstellt. Aufgrund des erstellten Modells kann das System seine Aufgabenstellungen anpassen und Erklärungen, Hinweise sowie - je nach Bedarf - Beispiele geben. Der Programmablauf wird dabei durch eine Software gesteuert und kann nur durch Antworten beeinflusst werden.

Ein Intelligentes Tutoren-System besteht dabei aus vier Komponenten:

- Das **Expertenmodul** versucht, das Fachwissen eines Lehrers abzubilden und beinhaltet Wissen über jenes Thema, welches das ITS (Intelligentes Tutoren-System) lehren soll.
- Das **Studentenmodul** repräsentiert das, was der Benutzer weiß oder auch nicht weiß. Dieses Wissen lässt das ITS erkennen, *wen* es unterrichtet, sodass es sich auf den Kenntnisstand des Benutzers einstellen kann.
- Das **Unterrichtsmodul** wählt den Inhalt und die Ordnung der Information und bestimmt Fragen, die es dem Benutzer als nächstes stellt.
- Das **Kommunikationsmodul** bestimmt die Kommunikationsform zwischen dem ITS und dem Lernenden. Die Verständigung kann zum Beispiel durch natürliche Sprache oder durch Menüführung erfolgen.

Im Gegensatz zu Büchern und einfachen tutoriellen Systemen kann ein Intelligentes Tutoren-System auch auf unvorhergesehene Aufgaben und auftretende Probleme eingehen bzw. individuelle Kritik üben. Weiters ist es möglich, auf den jeweiligen Benutzer angepasste Bereiche zu berechnen; der Lernende kann dabei einen entsprechenden Detaillierungsgrad auswählen, sich nicht verstandene Bereiche ausführlicher erklären und sein Wissen testen lassen.

Obwohl schon lange auf diesem Gebiet geforscht wird, werden ITS-spezifische Anwendungen nur selten in der Unterrichtspraxis eingesetzt. Gründe dafür sind einerseits die hohen Kosten, lange Entwicklungszeiten, aufwendige Einführung und Wartung; andererseits gibt es viele unterschiedliche Forschungsinteressen von Bildungspsychologen und Informatikern. Auch fehlen ausgetestete Prototypen und Marktprodukte. Dennoch gibt es einige wenige ITS, welche von der U.S. Navy und

der Air-Force in Auftrag gegeben wurden, mit dem vorrangigen Ziel, Offiziere und Technikpersonal zu trainieren.

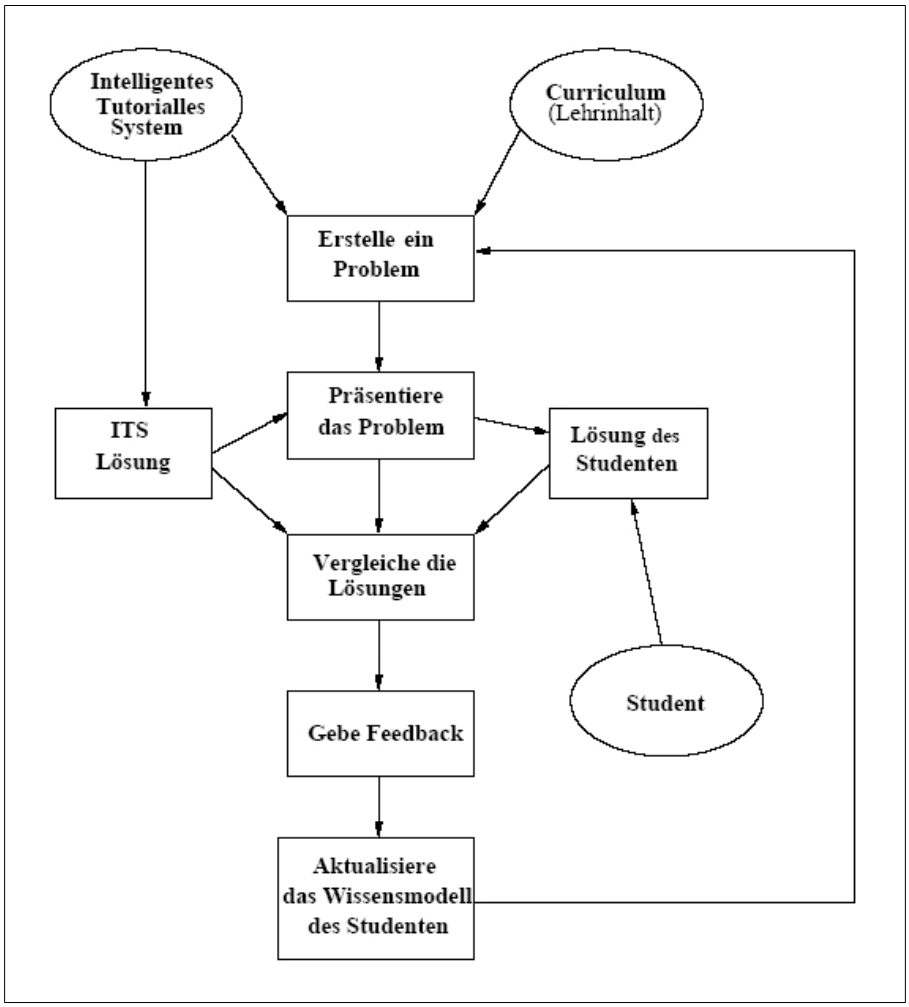


Abb. 2.4: Funktionsweise ITS [Nie03]

2.4.3 Persönliches Tutoring

Neben einem rein programmierten Tutoring, bei welchem die menschliche Komponente vernachlässigt wird, gibt es auch das traditionellere, **persönliche Online-Tutoring**.

Im besten Fall bilden sich Kleingruppen, welche von einem Tutor betreut werden. Um hier eine erfolgreiche Wissenskommunikation zu gewährleisten, muss der Tutor verschiedene Voraussetzungen und Funktionen erfüllen. Die Grundanforderungen an Tutoren in *face-to-face-Szenarien* oder für E-Tutoren sind dabei identisch:

- Der Tutor sollte eher begleitend und anregend agieren und die Verantwortung für den Wissenserwerb den Lernenden überlassen. Tutoring ist also primär beratend, grundlegende didaktische Kompetenzen sind dennoch eine Voraussetzung für diese Art der Lernbetreuung.
- Für eine beratende Tätigkeit im Bereich der Aus- und Weiterbildung sind Kenntnisse über Grundlagen und Methoden des selbstgesteuerten Lernens notwendig.
  Die Hauptaufgabe des Tutors ist die Förderung kooperativen und kollaborativen Verhaltens; dies erfordert die Kompetenz, soziale Prozesse in einer Gruppe wahrzunehmen, zu verstehen und entsprechend zu reagieren.
- Nicht nur die Fähigkeit, sich in die soziale Lage der Lernenden hineinzuversetzen, ist wichtig, sondern auch ein Hineinversetzen in die intellektuelle Situation einer Lerngruppe. Der Tutor muss verstehen, welche kognitiven Prozesse ablaufen und diese gegebenenfalls in Bezug zu sozialen Prozessen setzen.
- Neben dem Erkennen problematischer Situationen ist auch das Handeln und dessen Art von Bedeutung. Als Tutor sollte man über ein ausreichendes Repertoire an Kommunikations- und Moderationskompetenzen verfügen.

2.4.4 Programmiertes versus persönliches Tutoring

Zur Abwägung der Vor- bzw. Nachteile von programmiertem und persönlichem Tutoring im praktischen Einsatz wurde an den Euro-Schulen Hannover[2] die Einführung eines E-Learning-Systems getestet. Hier bot sich ein breites Kundenspektrum und ein breites Einsatzfeld für ein E-Learning-Modell. Besonders Wert wurde auf das begleitende Tutoring-Modell genommen, wobei vorrangig ein so genanntes Lerntyp[3]-bezogenes Tutoring statt fand. Diese Art des Tutorings begleitet den

[2] http://www.euro-schulen-hannover.de

[3] Im Prinzip ergeben sich (je nach Definition) bei einer groben Einteilung drei Lern-Typen: visueller, auditiver und handlungsorientierter Lerntyp.

Lernenden von Anfang an, entwickelt zusammen Lernpläne und -ziele und kann auch selbständig eingreifen.
Bei dem verwendeten Modell wird der gesamte Lernzyklus tutoriell begleitet und beginnt beim ersten Eintritt eines Lernenden in das Lernforum.

Die **erste Phase** besteht in der Ermittlung der Voraussetzungen und Fähigkeiten im Umgang mit Lernsoftware. Weiters werden Lernplan und -ziel erstellt und im E-Learning-System dokumentiert. Nach Erhalt der Zugangsdaten muss sich der Lernende mittels einer integrierten Korrespondenzfunktion im System anmelden. Bei Nichtanmelden wird der Tutor automatisch aktiv und spricht die Lernenden per E-Mail an.

Die **zweite Phase** besteht für den Tutor hauptsächlich aus einem Beobachten der Lernschritte und Beantworten von Fragen, die per Korrespondenz oder direkt im betreuten Lernraum gestellt werden. Weiters werden aus der Lernbeobachtung bzw. Diskussion mit den Lernenden weiterer Wissens-, Übungs- und Gesprächsbedarf ermittelt. Der Tutor erstellt zusätzliche (WBT-)Texte und stellt Fragen bzw. Aufgaben in den betreuten Lernraum.
Die Lernenden beantworten diese mit Lösungen oder Kommentaren bei ihrer nächsten Sitzung, oder der Tutor meldet sich beim Lernenden per Korrespondenz, um auf die Probleme im Lernzyklus einzugehen. Handelt es sich stärker um einen handlungsorientierten Lerntyp, werden zusätzliche Aufgaben als Dateien im Arbeitsbereich zum Download hinterlegt, die dann offline bearbeitet werden können.

Online-Meetings werden zu regelmäßigen Terminen abgehalten, bei denen der Tutor die Diskussion im Chat moderiert. Wichtig ist hier, dass es nicht zu einem reinen Frage-Antwort-Spiel zwischen einem Lernenden und dem Tutor kommt, sondern sich die ganze Gruppe daran beteiligt.

Bei dem vorgestellten System ist die Lernkontrolle ein Meilenstein im Lernzyklus. Den Zyklus begleitend können Lernende noch Zusatzinformation aus Beiträgen und Literatur gewinnen; auch ist die Suche in einem KMS (Knowledge Management System) möglich und es stehen noch weitere, hier nicht erwähnte, Tools zur Verfügung.

Zusammenfassend kam man zur Erkenntnis, dass es nicht optimal ist, sich nur für ein rein programmiertes oder rein persönliches Tutoring bei der Entwicklung eines komplexeren E-Learning-Konzeptes zu entscheiden. Im Prinzip ist es immer

eine Mischform von beiden Tutoring-Typen, welche ein E-Learning-System einer breiten Schicht von Lerntypen attraktiv macht. In welche Tiefe die Aufgaben eines programmierten Tutors gehen, hängt stark von der Einschätzung über die Zielgruppe und der damit verbundenen Lerntypen ab.

2.5 Internationale E-Learning-Gremien und -Initiativen

Der Herstellermarkt für Lernplattformen und Autorensysteme ist sehr groß und dynamisch, wodurch die Einhaltung internationaler E-Learning-Standards immer wichtiger wird; auch ist dadurch das Kriterium der Interoperabilität gegeben, womit sich erst ein Erstellen von beispielsweise sehr kostenintensiven WBTs rentiert.

In den vergangenen Jahren haben sich in den USA bzw. Europa vermehrt mehrere Standardisierungskonsortien gebildet, die offene Technologie-Standards zur Interoperabilität von Lernplattformen, Autorensystemen und WBTs definieren. Es besteht inzwischen ein recht kompliziertes Netzwerk der verschiedenen Gremien, welche ihre Arbeitsergebnisse austauschen und durch das IEEE (Institute of Electrical and Electronics Engineers) LTSC (Learning Technology Standards Committee) standardisieren lassen.
Momentan gibt es 15 verschiedene Arbeitsgruppen, die sich mit Teilaspekten des E-Learnings befassen. Vorschläge werden an die Untergruppe SC36 (Standards for Information Technology for Learning, Education and Training), ein Zusammenschluss der ISO (International Organization for Standardization) und der IEC (International Electronical Commission), weitergeleitet und zum weltweiten Standard erklärt.

2.5.1 Standardisierungsgremien und Standards

Zum besseren Verständnis des Kapitels erfolgt zuvor eine kurze Definition der Begriffe Standard und Spezifikation:

Standard Ein Standard stellt eine bestimmte Mindesteigenschaft von Produkten, Methoden oder Abläufen sicher und dient der Vereinfachung von

Entwicklungs- und Anwendungsprozessen. Unterschieden wird zwischen zwei Arten: *De-jure*-Standards und *De-facto*-Standards.

Nationale und internationale Standardisierungsgremien können Standards zertifizieren; Standards versehen mit einem Zertifikat werden auch De-jure-Standards oder Norm genannt. De-facto-Standards gelten im Prinzip als Standard, gehören aber nicht zu den formalen und offiziell akkreditierten Standards.

Spezifikation wird in diesem Zusammenhang verwendet, wenn detaillierte Vorschläge und Richtlinien noch nicht von offiziellen Standardisierungsgremien zertifiziert wurden. Die Mehrheit der Arbeitsergebnisse der Gremien, die sich mit E-Learning beschäftigen, werden als Spezifikationen herausgegeben und erst zu einem späteren Zeitpunkt zu einem Standard erklärt. Zu den Standardisierungsgremien zählen:

- Deutsches Institut für Normung (DIN)[4]
- American National Standards Institute (ANSI)[5]
- Institute of Electrical and Electronics Engineers (IEEE)[6]
- International Organization for Standardization (ISO)[7]
- National Institute of Standards and Technology (NISO)[8]

Im Folgenden wird ein Überblick (siehe Abb. 2.5 auf der nächsten Seite) über die wichtigsten Standardisierungsgremien und veröffentlichten Standards gegeben:

AICC (Aviation Industry Computer Based Training Commitee)[9] ist ein internationaler Verband von Fachleuten, welche sich mit E-Learning-gestützten Kursen beschäftigen. Vorrangig werden Richtlinien für die Entwicklung, Bereitstellung und Bewertung von Kurssystemen erarbeitet. AICC ist für weitverbreitete und akzeptierte Interoperabilitätsstandards für Kurse (online und offline) verantwortlich.

Nach außen hin repräsentiert sich das AICC über drei Dokumentgruppen:

[4] http://www2.din.de
[5] http://www.ansi.org
[6] http://www.ieee.org
[7] http://www.iso.org
[8] http://www.niso.org
[9] http://www.aicc.org

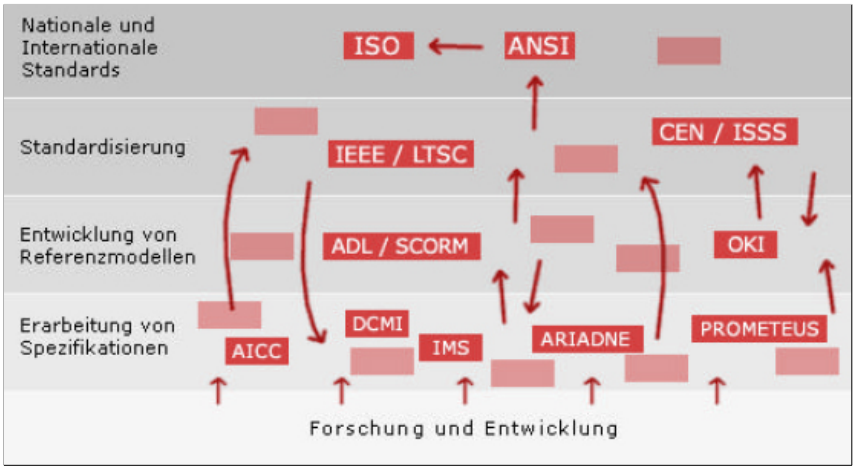


Abb. 2.5: Kooperationsnetzwerk der Standardisierungsgremien [ano05]

Richtlinien und Empfehlungen, kurz AGRs (AICC Guidelines and Recommendations), technische Reports und Arbeitsdokumente. Kurse, welche den AGRs folgen, sind hierarchisch aufgebaut, beginnend mit einer Wurzel, dem Kurselement. Die Wurzel beinhaltet *Blocks* und *Objectives*, so genannte Assignable Units bilden dabei die kleinste Einheit. Assignable Units können zu Blöcken zusammengefasst und mit Bedingungen verknüpft werden. Beschrieben werden die genannten Elemente mittels unterschiedlicher ASCII-Dateien. Für ein funktionierendes Zusammenwirken der einzelnen Dateien, wird jedem Element eine ID vergeben.

Die Vorteile des AICC liegen in der weiten Verbreitung und angebotenen Zertifizierung von Produkten. Als Nachteil stellt sich das veraltete und umständlich zu implementierende Dateisystem dar und ein Fehlen standardisierter Elemente für Metadaten.

ADL (Advanced Distributed Learning)[10] ist eine Initiative zur Entwicklung von Standards zur Nutzung von Lern- und Informationstechnologien bei Ausbildung und Training und zur Sicherung des Zugriffs auf hochqualitative Lernmaterialien.

SCORM (Sharable Content Object Reference Model)[11] ist eine von ADL entwickelte Empfehlung zur Standardisierung von Lernobjekten. Damit soll unter Berücksichtigung von Anforderungen und Lösungsvorschlä-

[10] http://www.adlnet.org
[11] http://www.adlnet.org/scorm

ge aus der Praxis ein universales Inhaltsmodell entwickelt werden (siehe Kapitel 3).

ARIADNE (Alliance of Remote Instructural Authoring and Distribution Networks for Europe)[12] ist ein Projekt der Europäischen Union und der Schweizer Regierung. ARIADNE I & II fand in zwei Phasen zwischen 1996 und 2000 im Rahmen des Programmes „Telematics for Education and Training“ innerhalb des vierten Rahmenprogramms der Europäischen Union statt.

Ziel des Projektes war die Förderung der gemeinsamen Nutzung und Wiederverwendung von pädagogischen Materialien und Werkzeugen (Authoring Tools, Core Tools) zur Herstellung, Indexierung, Verwaltung und Zugänglichkeit solcher Dokumente. Ein Netzwerk von Gruppen, die verwandte Wissensgebiete bearbeiteten, diente als Basis und Wissensspeicher, auf dessen Grundlage Lernprogramme erstellt und vertrieben werden konnten. Am Projekt waren über 30 Partner beteiligt.

ARIADNE II verfolgte zwei Ziele: Validierung der technischen Resultate von ARIADNE I mit einer grossen Anzahl von Partnern und Verbesserung der Werkzeuge von ARIADNE durch die Schaffung neuer Verbindungen im Netzwerk und Überarbeitung der Werkzeuge. Ein Teil der Arbeit galt auch der Verbreitung und Information, der Standardisierung und der Unterstützung von Aktivitäten von Benutzern, die nicht am Projekt beteiligt waren (*ARIADNE User Group*). Das Projekt wurde Ende Juni 2000 abgeschlossen.

Um das Weiterwirken der Erfahrungen und Errungenschaften des Projektes sicherzustellen, wurde die *Fondation ARIADNE*[13] ins Leben gerufen. Diese Stiftung widmet sich der weiteren Entwicklung der ARIADNE-Werkzeuge, beteiligt sich an den Arbeiten zur Standardisierung der Meta-Daten der Lernobjekte - LOM (Learning Objects Metadata) - im Rahmen des IEEE, organisiert Seminare und internationale Konferenzen und gibt Bulletins heraus.

CEN (European Committee for Standardization)/ISSS (Information Society Standardization System)[14] ist ein europäisches Normierungsgremium mit dem Ziel, Empfehlungen auf Basis einer Zusammenarbeit mit anderen Organisationen zu erarbeiten.

[12] http://ariadne.unil.ch
[13] http://www.ariadne-eu.org
[14] http://www.cenorm.be

Beispiele hierfür sind:

- Internationalisierung von LOM
- Mehrsprachigkeit von LOM
- Definition von Sprachkompetenzen
- Intellectual Property/Copyright
- Qualitätssicherung
- EML (Educational Modeling Language).

Geachtet wird dabei stets auf europäische Forderungen wie beispielsweise die Multikulturität und Mehrsprachigkeit.

DCMI (Dublin Core Meta Initiative)[15] wurde im Jahre 1995 durch das W3C mit dem Ziel ins Leben gerufen, einen einheitlichen Metadaten-Standard zu erstellen, wobei ein internationaler und interdisziplinärer Konsens über die zu verwendenden Elemente angestrebt wurde, um eine möglichst breite Anwendbarkeit zu gewährleisten.

Die DCMI verfolgt mit der Metadaten-Spezifikation vier Hauptziele:

- einfache Erstellung und Pflege von Metadaten
- einfache Semantik der Metadaten
- internationale Ausrichtung
- mögliche Erweiterbarkeit

Die Spezifikation, das so genannte DCMES (Dublin Core Metadata Element Set), besteht aus 15 Kernelementen, die den drei Gruppen Content, Intellectual Property und Instantiation zugeteilt werden können. Elemente des DCMES werden durch Qualifikatoren spezifiziert, die wiederum in zwei Klassen eingeteilt werden können.
Anwendung findet der Standard bei Archivsystemen, Bibliotheken und auch beispielsweise zur Kodierung in HTML oder RDF/XML; bezüglich Lernsoftware sind die Vorteile eher gering.

IEEE LTSC (Learning Technology Standards Committee)[16] nimmt mit dem LTSC bei der Entwicklung von Standards eine internationale Vorreiterrolle ein und besteht aus fünfzehn Arbeitsgruppen (Working Groups), die in fünf

[15] http://dublincore.org
[16] http://ltsc.ieee.org

Kategorien zusammengefasst werden können (General, Learner-related, Content-related, Data and Metadata, Management Systems and Applications).

Im Rahmen der LTSC-Arbeiten können vier Spezifikationen identifiziert werden:

- **LTSA (Learning Technology System Architecture)** beschreibt eine abstrakte Systemarchitektur von Lerntechnologien. Die Ziele bestehen aus der Definition einer einheitlichen Bezeichnungssystematik (*Notation*) für Lehr- und Lernumgebungen, der Unabhängigkeit von pädagogischen Zielstellungen, Inhalten und technischen Plattformen und einer Verbesserung der Analyse und des Vergleichs von Lehr- und Lernumgebungen.
- **LOM (Learning Objects Metadata)** beschreibt digitale und nicht-digitale Lernmodule (siehe Unterunterabschnitt 3.2.1.3) und ermöglicht somit die Identifikation und den Austausch von Modulen. LOM liefert eine Beschreibungsmöglichkeit des Inhalts in Form von Metadaten(-schemata), welche bei einer Kurskonstruktion verwendet werden können.
  Der Schwerpunkt der Beschreibung liegt dabei auf der Inhaltsbeschreibung und nicht auf der Inhaltsvermittlung.
- **PAPI (Public and Privat Information)** bezieht sich auf die Daten der Lernenden (persönliche Informationen, Informationen über das Lernverhalten und die Lernleistungen, über Arbeiten und Abschlüsse der Lernenden, sowie über Benutzerpräferenzen). Informationen sollen zwischen Lernumgebungen ausgetauscht werden und dienen dabei einer Personalisierung und/oder Effizienzsteigerung des Lernprozesses (Adaptoring/Tutoring).
- **CMI (Computer Managed Instruction)** beschäftigt sich mit Spezifikationen für den Austausch, die Kombination und Administration von Kursen. Der Standard umfasst umfangreiche Informationen für die systemplattform- und applikationsunabhängige Integration von WBTs in Learning-Management-Systemen.

ISO/IEC JTC (Joint Technical Committees) SC36 (Standards for Information Technology for Learning, Education and Training)[17] bilden vereinigte Gremien (JTC) zur Erarbeitung von Standards. Der Untersuchungsausschuss SC36

[17] http://jtc1sc36.org

beschäftigt sich hauptsächlich mit der Standardisierung auf dem Gebiet der Informationstechnik zur Unterstützung der Automatisierung von Lernen, Bildungseinrichtungen und Lernressourcen. Im Vordergrund stehen auch hier Interoperabilität und Wiederverwendbarkeit der Ressourcen und Systeme auf technischer Ebene. Inhaltliche Gesichtspunkte werden dabei nicht betrachtet.

IMS Global Learning Consortium[18] ist eine gemeinnützige Gesellschaft, die aus einem internationaler Zusammenschluss unterschiedlicher Bildungs- und Regierungsorganisationen entstanden ist. Der Gesellschaft gehören auch Hersteller und Nutzer von diversen CBTs, WBTs und E-Learning-Plattformen an.

Der vom Konsortium entwickelte Standard IMS (Instructional Management System) definiert eine Datenstruktur für den Datenaustausch, der Suche und Wiederverwendbarkeit von (digitalen) E-Learning-Inhalten. Die Realisierung erfolgt mittels einem so genannten *Content Package* (siehe Abschnitt 3.2.1.2 auf Seite 42), das primär alles enthalten und zusammenfassen soll, was für die jeweilige Lerneinheit benötigt wird.
Das Package besteht aus zwei Teilen, einem XML-Dokument (*Manifest*), das Aufbau und Inhalt des Paketes beschreibt (siehe Abschnitt 3.2.1.2 auf Seite 42) und den Dateien, auf die im Manifest referenziert wird.

Der große Vorteil bei IMS liegt in der schon frühen Verwendung von XML und engen Zusammenarbeit mit SCORM, nachteilig ist das Fehlen einer Laufzeitumgebung.

PROMETEUS (Promoting Multimedia Access to Education and Training in the European Society)[19] wurde von der Europäischen Kommission mit der Absicht gegründet, den multimedialen Zugang zur Aus- und Weiterbildung in der europäischen Gesellschaft zu fördern.

Die Ziele der Initiative, dargelegt in einer Absichtserklärung, sind eine Erhöhung der Effizienz bei Zusammenarbeit zwischen Behörden und Einrichtungen der Aus- und Weiterbildung sowie zwischen Anwendern und Anbietern von Lernsoftware. Weiters soll die Entwicklung gemeinsamer europäischer und internationaler Normen für digitale multimediale Lerninhalte und -dienste gefördert werden. Weiters soll die Schaffung neuer Lehr- und Aus-

[18] http://www.imsproject.org
[19] http://www.prometeus.org

bildungsmethoden und neuer Lernumgebungen unterstützt werden. Dabei wird großer Wert auf erschwingliche Lösungen und Plattformen, die auf offenen Standards und den besten Verfahren basieren, gelegt. Informationssysteme sollen öffentlich zugänglich und miteinander verbindbar sein.
Der Zusammenarbeit soll eine globale Dimension verliehen werden; es müssen jedoch bestimmte gemeinsame Leitlinien für die Organisation der künftigen Zusammenarbeit befolgt werden.

EML (Educational Modeling Language)[20]/OUNL (Open University Niederlanden)[21] basiert auf einem Metamodell zur pädagogischen Modellierung von Lernumgebungen; dabei hat die Einbettung von Lernobjekten in einen didaktischen Kontext zentrale Bedeutung. Das Modell besteht aus vier Komponenten: Theories of Learning and Instruction, Learning Model, Domain Model, Units of Study Model.

Beschrieben wird hauptsächlich der didaktisch kommentierte Aufbau von Lehrinhalten und nicht die zugrundeliegende Architektur. Weiters werden Beziehungen und Interaktionen zwischen den Komponenten der Lernsituation beschrieben und auf die Lerninhalte abgebildet, was wiederum eine Anpassung des Materials an die Bedürfnisse des jeweiligen Lerners erlaubt. Die Umsetzung erfolgt mittels XML-Schemata, wo sowohl Wissenseinheiten selbst als auch zugehörige Metadaten abgebildet werden können.

OKI (The Open Knowledge Initiative)[22] wurde vom MIT (Massachusetts Institute of Technology)[23] zusammen mit neun namhaften amerikanischen Universitäten initiiert und versucht, in der Entwicklung von Lernsoftware und Learning-Management-Systemen eine offene, erweiterbare Architektur für die Hochschulbildung zur Verfügung zu stellen - sowohl für kommerzielle Produktanbieter als auch für Hochschul-Produktentwickler. Sie soll als standardisierte und modularisierte Basis mit in Java programmierten Anwendungsschnittstellen für die Entwicklung von Lernplattformen dienen und Teil globaler Normierungsbestrebungen im E-Learning sein.

Ziel ist es, eine Schaffung von Produkten mit spezifischeren Anforderungen als es kommerzielle Software (z.B. WebCT[24], siehe Abschnitt 4.2.2.3 auf Sei-

[20] http://eml.ou.nl
[21] http://www.ou.nl
[22] http://web.mit.edu/oki
[23] http://www.mit.edu
[24] http://www.webct.com

te 92) bietet, um teure lokale Eigenentwicklungen zu vermeiden und internationale Austauschbarkeit zu gewährleisten. Die wichtigsten Bereiche sind dabei die Kurs-Administration, Inhalts-Aufbewahrung, Bewertungs-Module sowie Kommunikationsdienstleistungen.

GEM (Gateway to Educational Materials)[25] ist eine Initiative des ERIC/IT (Education Resources Information Center on Information Technology) mit dem Ziel, das Auffind- und Verwertbarkeitsdefizit zu beheben und einen Zugang (*Gateway*) zu einer qualitativ hochwertigen Auswahl von Lehr- und Lernmaterial bereitzustellen. Dadurch soll besonders Lehrenden das Finden von relevantem Lehrmaterial erleichtert werden. Der Zugang erfolgt mittels Listen, die beispielsweise nach Oberbegriff, Schlüsselwörtern (*Keywords*) und Ausbildungsstufe (Jahrgangsstufe) geordnet sind und direkt aus dem Gateway heraus zugänglich sind. Das Dublin Core Metadata Element Set wurde als Basis für die GEM-Elemente gewählt.

Microsoft LRN (Learning Resource Interchange) : Im Jahr 2000 begann Microsoft selbst die IMS Content Packaging 1.1 und IMS Metadata 1.2 - Spezifikationen auf kommerzieller Basis zu implementieren; SCORM 1.2 und dessen Run-Time Environment wurden dabei ebenfalls unterstützt.

Die Spezifikation ist ein XML-basiertes, erweiterbares Schema, das eine einheitliche Erstellung und Verwaltung von Lerninhalten in einem Standardformat sicherstellt; geltend nicht nur für webbasierte Lernformen sondern auch andere Medien wie beispielsweise CD-ROM. Gemeinsam mit dem Schema wurde auch das „LRN Toolkit“ publiziert; es stellte Werkzeuge (Editor, Generatior, Validator, Konverter) und Beispieldateien im LRN-Format zur Verfügung.

Entwickelt wurde LRN bis zur Version 3.0. Nachdem aber der IMS Standard immer mehr in das SCORM Modell übernommen wurde und Microsoft bei einer Weiterentwicklung von LRN eine noch größere Unübersichtlichkeit innerhalb der E-Learning Standards sah, wurde die Entwicklung beendet und das Toolkit nicht mehr angeboten.

CanCore[26] **:** Im Jahr 2000 wurde die CanCore Metadaten Initiative als Diskussions- und Umsetzungsplattform zum Thema Metadaten für kanadische E-Learning-Projekte gegründet. Ziel von CanCore ist die Förderung der

[25] http://www.thegateway.org
[26] http://www.cancore.ca/en/

korrekten und erfolgreichen Erfassung von richtigen und präzisen Metadaten im E-Learning. Die Initiative widmet sich insbesondere der Umsetzung des LOM-Standards und gibt Beispiele, wie LOM umgesetzt werden kann. Außerdem werden Empfehlungen und Vorschläge zur Verwendung der LOM-Elemente bezüglich Syntax und Semantik erarbeitet.

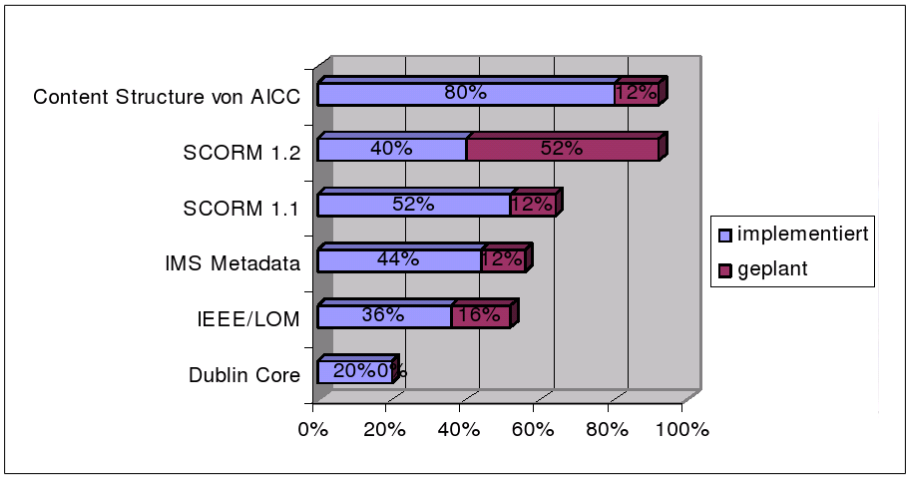


Abb. 2.6: Unterstützte Standards in LMS [Gel05]

2.6 Fazit/Ausblick

Mit Einzug des E-Learning und den damit verbundenen technischen Möglichkeiten verlieren das traditionelle Lernen und Lehren immer mehr an Bedeutung, eine teilweise Automatisierung wird angestrebt, indem Tutoring-Systeme zum Einsatz kommen. Es sollen einerseits Informationen zur Verfügung gestellt und andererseits Kommunikation ermöglicht werden. Bei vielen Projekten wird jedoch der zur Realiserung benötigte Aufwand oftmals unterschätzt. Schon bei einfacheren tutoriellen Systemen sind eine hohe Entwicklungszeit und hohe Kosten gegeben, komplexe Systeme wie ITS sind teilweise nicht realisierbar bzw. unausgereift.
Eine vollkommene Automatisierung (programmiertes Tutoring), vielmals angestrebt, wird nur schwer durchzusetzen sein, da von vielen noch eine face-to-face-Lehre bzw. ein persönliches Tutoring bevorzugt wird und es oftmals einer adäquaten Unterstützung durch Lehrpersonal bei der Begleitung von E-Learning-Prozessen bedarf.

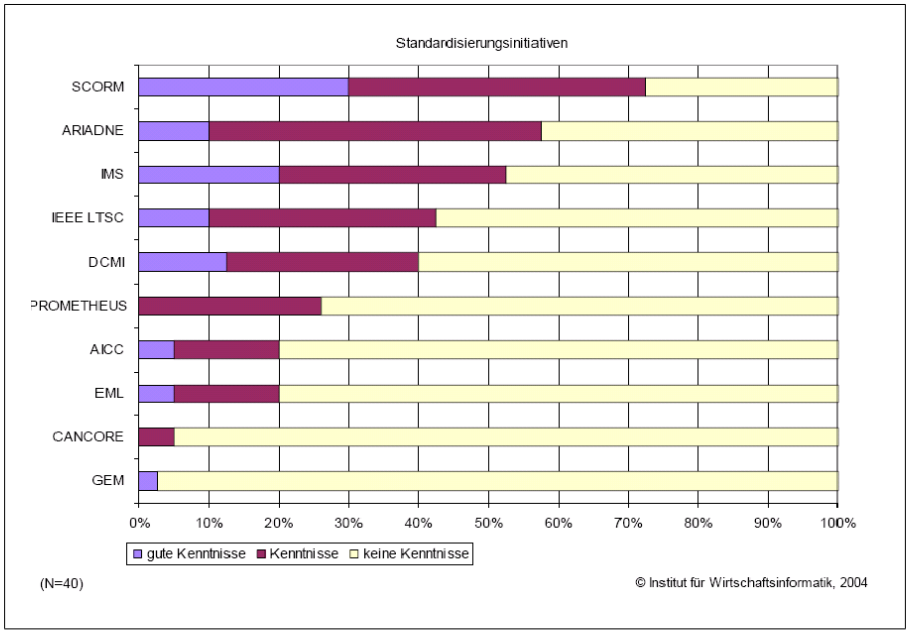


Abb. 2.7: Bekanntheitsgrad von Standardisierungsinitiativen [Gel05]

Inzwischen hat sich aber auch bereits ein breites Spektrum an entsprechenden Spezifikationen, den E-Learning-Content betreffend, gebildet. Teilweise werden diese bereits erfolgreich in Produkten angewandt, andere haben sich bereits als Industriestandards etabliert – ein prominenter Vertreter davon ist SCORM (siehe Kapitel 3 auf der nächsten Seite).

Trotz der Vorteile bei der Verwendung von Standards, gibt es auch viele nachteilige Effekte. Der Aufwand bei der Anwendung von Spezifikationen und Standards ist teilweise recht hoch - oftmals ist eine Produktneuentwicklung notwendig. Dennoch ist eine positive Entwicklung erkennbar, da die meisten Firmen eine Unterstützung von Standards anstreben, auch wenn zur Zeit die Qualität und Robustheit entsprechender Implementierungen noch unterschiedlich sind.

Literatur: [Kai01], [Pet04], [Rie02], [Häf03], [Rau03], [Küp02], [Zum02], [Koc02], [Mon04], [Exn05], [Rau01], [Han02], [MHH02], [Wik], [MFS03], [Blu98], [Nie03], [Gel05]

Kapitel 3

SCORM 2004

> Among the many upcoming standards for e-learning, it turns out, that the IMS and SCORM standards suite are widely accepted in the domain of higher education. The most important parts are IMS/SCORM-CP, SCORM-RTI and IMS-QTI. The main purpose of using standards is reusability and interoperability.
>
> (Monnard/Brugger 2003)

Im folgenden Kapitel wird das von ADL entwickelte Referenzmodell SCORM in seinen Details und aktuellsten Version vorgestellt. Es wird kurz die Entstehungsgeschichte umrissen, um dann genauer die Architektur zu erläutern. Nachdem das so genannte Sequencing and Navigation - die Ablaufsteuerung von Kursen - zentraler Bestandteil von SCORM 2004 ist, wird es am Ende des Kapitels ausführlicher behandelt. Abschließend werden etwaige Einschränkungen diskutiert und ein Ausblick in zukünftige Entwicklungen gegeben.

3.1 Einführung

3.1.1 ADL Initiative

Die Advanced Distributed Learning (ADL) Initiative wurde 1997 vom amerikanischen Verteidigungsministerium [1] und dem Ministerium für Wissenschaft und Technologie [2] gegründet. Vorrangiges Ziel war es, Arbeitsergebnisse bestimmter

[1] Department of Defense (DoD)
[2] White House Office of Science and Technology Policy (OSTP)

Organisationen, die im Bereich der Standardisierung von E-Learning tätig sind, in einem gemeinsamen Referenzmodell, dem Sharable Content Object Reference Model, zu vereinen. Folgende Standardisierungsgruppen wurden dabei eingebunden: AICC (Aviation Industry Computer Based Training Commitee), DCMI (Dublin Core Meta Initiative), IEEE (Institute of Electrical and Electronics Engineers), Instructional Management System (IMS) Global Learning Consortium und als einzige nicht-amerikanische Institution die ARIADNE Foundation.

Die Aufgabe von ADL besteht nun darin, einen sicheren Zugang zu qualitativ hochwertiger Bildung und Training, die auf individuelle Bedürfnisse ausgerichtet sind, zu gewährleisten. Kosteneffizienz sowie Unabhängigkeit von Zeit und Ort sind weitere Forderungen der Initiative. Besonderes Augenmerk wird dabei auf Netzwerk- und Web-basierende Kurse gelegt.

Neben der Zusammenführung und Vereinheitlichung von Spezifikationen waren folgende Anforderungen weitere Motive für das Referenzmodell:

Wiederverwendbarkeit (Reusability): Inhalte können in anderen Programmen und Kontexten ebenfalls verwendet werden. Neben der Wiederverwendbarkeit einzelner Inhalte ist diese auch für verschiedene Ausgabemedien von Bedeutung. SCORM ist zwar hauptsächlich ein Standard für Web-basierende Lerninhalte, eine Ausgabe auf andere Medien, beispielsweise CD-ROMs soll dennoch möglich sein. Dies erfordert aber eine Entkoppelung der Inhalte von proprietären LMS und Bereitstellung offener Schnittstellen.

Erreichbarkeit (Accessability): Inhalte sollen von verschiedenen Personen von unterschiedlichen Standorten aus abgerufen und benutzt werden können.

Beständigkeit (Durability): Eine Beständigkeit gegenüber Weiterentwicklung von Betriebssystemen und Software ohne kostenintensives Redesign, Rekonfiguration oder Neuprogrammierung soll eine Abwärtskompatibilität zu früheren SCORM-Versionen sichern.

Interoperabilität (Interoperability): Standard konforme Kurse müssen auf verschiedenen Plattformen unter verschiedensten E-Learning-Systemen funktionieren.

Erschwinglichkeit (Affordability): Es wird eine Steigerung von Effizienz und Produktivität bei gleichzeitiger Reduzierung des Zeit-/Kostenaufwands gefordert.

Anpassungsfähigkeit (Adaptibility): Es soll eine Möglichkeit geben, Instruktionen an individuelle oder organisationelle Gegebenheiten anzupassen.

Ein weiteres Ziel ist es, zukünftig durch den gemeinsamen Standard und Hinzufügen von Metadaten bei Lerneinheiten nach bestimmten Lerneinheiten zu suchen. Mittels SCORM sollen neuartige Suchmaschinen konzipiert werden, die nach Inhalten suchen können. ADL hat daher auf lange Sicht vor, wie in Abb. 3.1 verdeutlicht, die Suche jederzeit und auch im WWW zu ermöglichen, um sich in Echtzeit ein Lernprogramm zu einem beliebigen Thema generieren lassen zu können.

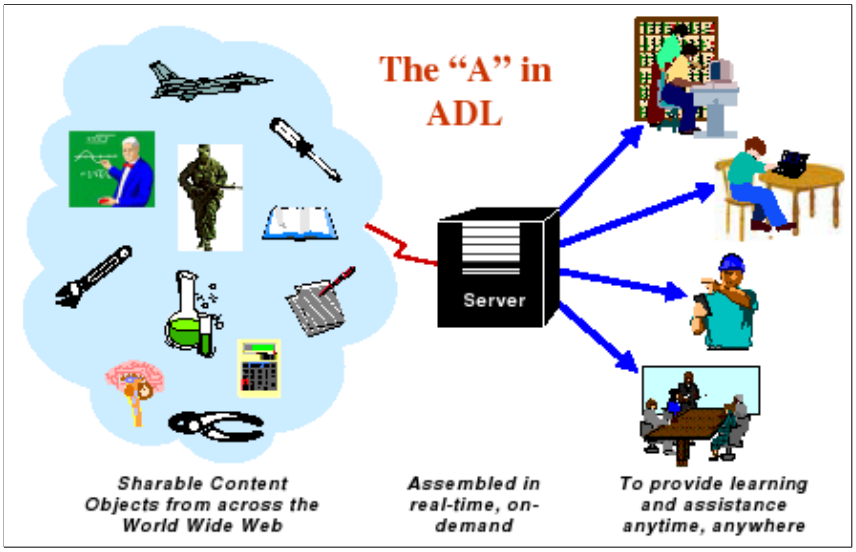

The "A" in ADL
Server
Sharable Content Objects from across the World Wide Web
Assembled in real-time, on-demand
To provide learning and assistance anytime, anywhere

Abb. 3.1: Die ADL-Vision [ADL04a]

3.1.2 SCORM

SCORM steht für Sharable Content Object Reference Model, folgende Spezifikationen und Standards wurden dabei vereinigt wurden:

- IEEE Data Model für die Kommunikation zwischen Inhaltsobjekt und LMS
- IEEE ECMAscript Application Programming Interface für die Kommunikation zwischen Inhalt und LMS

- IEEE Learning Objects Metadata (LOM) zum Auszeichnen der Daten mit Meta-Information
- IMS Content Packaging
- IMS Sequencing and Navigation

SCORM besteht aus vier Dokumenten, worin zusammengehörende und aneinander angepasste Sammlungen von Spezifikationen und Standards dargestellt werden; in der SCORM-Terminologie spricht man auch von Bücher (siehe Abb. 3.2 auf der nächsten Seite).

Overview: Im Überblicksdokument wird die Historie und die Vision von SCORM beschrieben sowie eine Einführung in die drei folgenden Teile und deren Zusammenhänge gegeben.

Content Aggregation Model (CAM): Das CAM beschreibt Ressourcen, die in Lernpaketen verwendet werden können, sowie die Möglichkeiten zur Zusammenfassung und Strukturierung von Ressourcen zu verteilbaren Lernpaketen. Als Ressourcen können insbesondere Dateien der unterschiedlichsten Formate oder URLs benutzt werden. Diese werden in Hierarchien organisiert, die als *Organisations* bezeichnet werden. Zusätzlich können zu einem derart strukturierten Lernpaket Metadaten spezifiziert werden, die anhand von Beschreibungen und Schlüsselworten eine gezielte Suche nach Lerninhalten ermöglichen. Das Gesamtpaket wird durch eine so genannte Manifestdatei im XML-Format definiert.

Run-Time Environment (RTE): Durch das RTE werden das laufzeitbezogene Verhalten eines LMS, die Schnittstelle vom LMS zu den in CAM spezifizierten Lernpaketen sowie die Verwendung von Benutzerdaten zur Laufzeit beschrieben. So können etwa Benutzer-Präferenzen gespeichert, Lernfortschritte festgehalten und Lernziele definiert werden.

Sequencing and Navigation (SN): Durch die SN-Spezifikation wird beschrieben, wie die Reihenfolge der Präsentation von Lerninhalten durch die Navigation des Benutzers variiert werden kann. Zu diesem Zweck werden so genannte Aktivitätsbäume (*Activity Trees*) beschrieben, die mögliche Reihenfolgen in Abhängigkeit von Benutzeraktionen definieren.

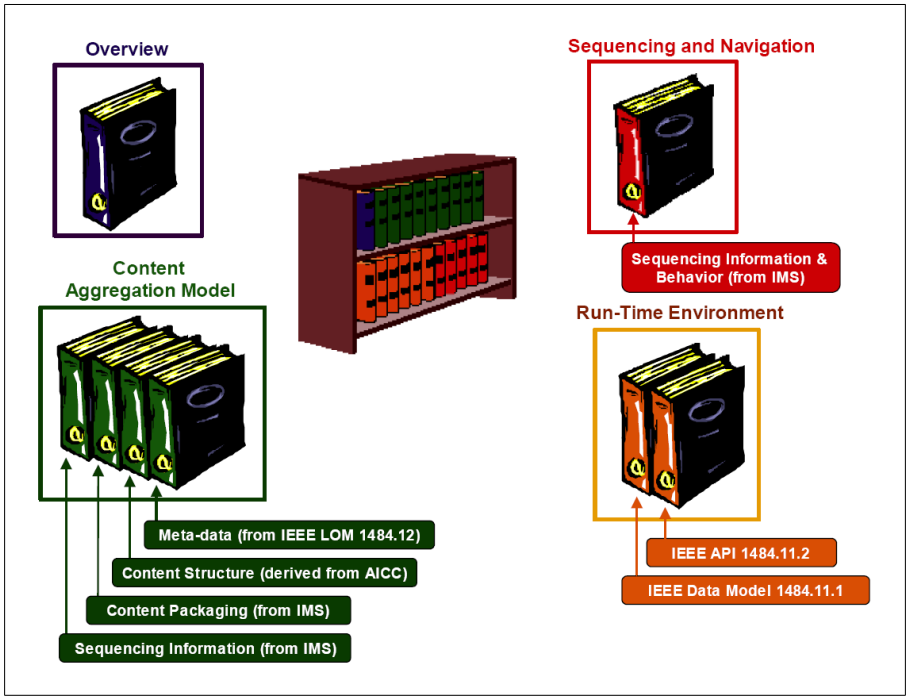


Abb. 3.2: Das SCORM-Bücherregal [ADL04a]

3.1.3 Geschichtliche Entwicklung

Die erste Version (1.0) von SCORM wurde im Jänner 2000 veröffentlicht noch unter dem Namen *Sharable Courseware Object Reference Model*. Erste Beispielimplementationen und Software für das Testen der Standardkonformität folgten. Bereits ein Jahr später wurde die Version 1.1 verabschiedet, in der Verbesserungsvorschläge und Korrekturen der Version 1.0 einbezogen wurden. Auch wurde die Testphase beendet und es kam zu einer Namensänderung. Damit wollte man verdeutlichen, dass SCORM nicht nur auf einen einzelnen Kurs sonder auf verschiedene Ebenen des Lerninhalts anwendbar ist; d.h. mit SCORM lassen sich kleine Einheiten (z.B. Lektionen) oder auch größere (z.B. Lehrpläne) beschreiben.

Im Oktober 2001 wurde von ADL die Version 1.2 veröffentlicht und es wurden das so genannte Content Packaging (siehe 3.2.1.2) eingeführt bzw. Metadaten (siehe 3.2.1.3) für die Beschreibung von Lerninhalten erweitert. Erstmals wurde SCORM hier in Bücher unterteilt, damals waren es noch drei, das vierte Sequencing and Navigation fehlte noch. In der nun aktuellste Version SCORM 2004 ist dieses Buch enthalten, auch die drei anderen Bücher wurden verbessert. Mit dieser Veröffent-

lichung wurde das Versions-Prinzip geändert; fortan wird jedes Buch unabhängig von den anderen weitergeführt und beginnt immer mit der Versionsnummer 1.4. Inzwischen ist bereits eine zweite Edition von SCORM 2004 erschienen, die Bücher tragen hier die Versionsnummer 1.3.1. Seit SCORM 2004 ist der Standard als stabil anzusehen. Jede weitere publizierte SCORM Version wird an technische Entwicklungen und Veränderungen an den zugrunde liegenden Spezifikationen angepasst.

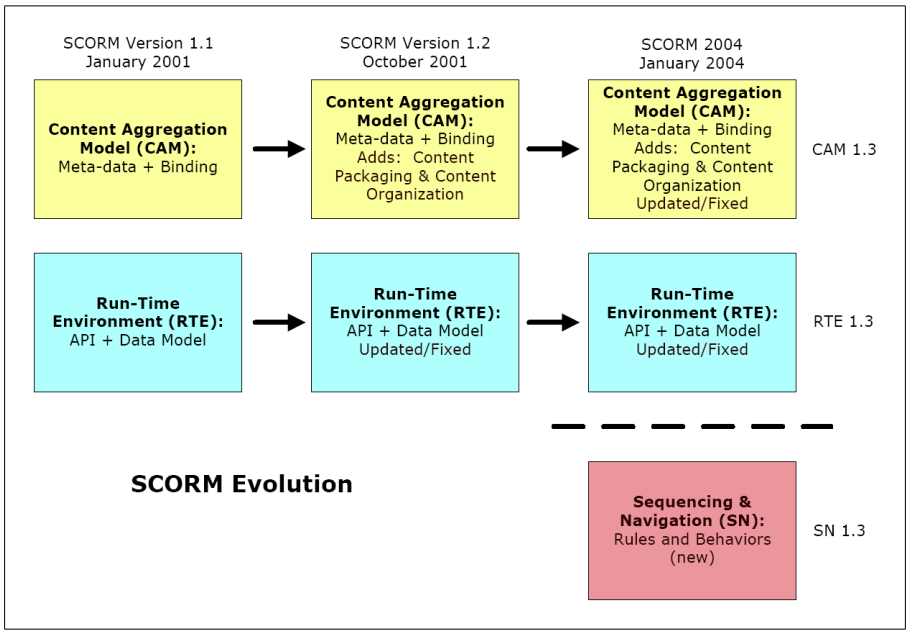


Abb. 3.3: SCORM Evolution [ADL04a]

3.2 Architektur

Die folgenden Abschnitte erläutern die vier bereits zuvor genannten Bücher.

3.2.1 Content Aggregation Model

Das CAM (Content Aggregation Model) ist ein Modell für die Zusammenstellung von Lern-Ressourcen zu komplexen Lernangeboten und beschreibt Komponenten

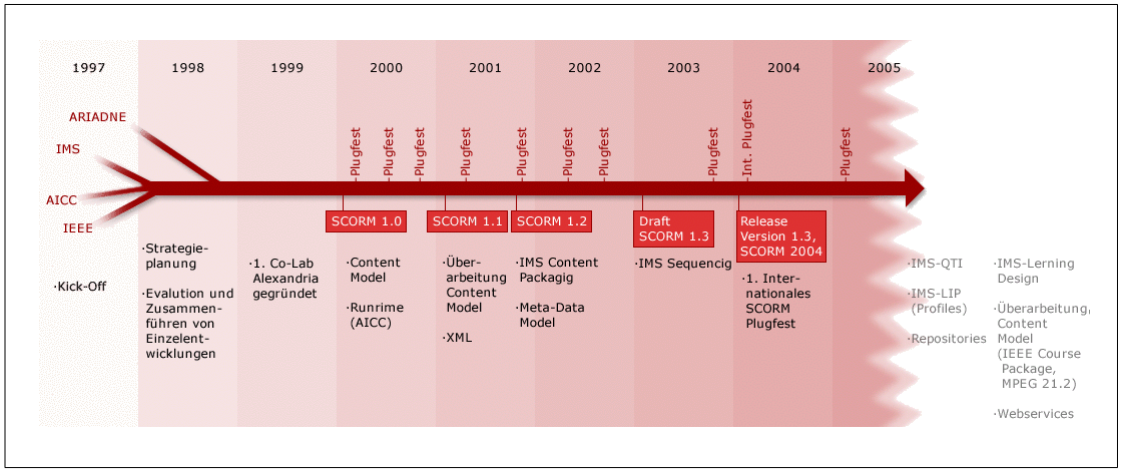


Abb. 3.4: Geschichtliche Entwicklung von SCORM [ano05]

aus denen sich ein Lernangebot zusammensetzt. Es gliedert sich in drei aufeinander aufbauenden Teile:

Content Model: Bezeichnet die einzelnen Lerninhalte

Content Packaging: Definiert Struktur und Verhalten (*Content Structure*) der Lernaktivität und vereinigt die Lern-Ressourcen zu Paketen.

Meta-Data: Ein Mechanismus der eine Beschreibung von Content Model-Instanzen ermöglicht

Sequencing and Navigation: Beschreibt, wie die Reihenfolge der Präsentation von Lerninhalten durch die Navigation des Benutzers variiert werden kann (siehe 3.3).

3.2.1.1 Content Model

Mit Hilfe des Content Models werden einzelne Lern-Ressourcen zu größeren Lerneinheiten zusammengestellt. Für diesen Zweck sind folgende Komponenten vorgesehen:

Assets: Ist die elementarste Form einer Lern-Ressource und repräsentiert in elektronischer Form jegliche Daten (siehe Abb. 3.5 auf der folgenden Seite), die an einen Web-Client übermittelt werden können; Beispiele hierfür sind: Texte, Bilder, Audiodateien oder Bewertungsobjekte (*Assessment Ob-*

jects). Mehrere einzelne Assets lassen sich wiederum zu einem neuen Asset bündeln. Weiters können Assets mit Metadaten ausgezeichnet werden, wodurch ein Suchen und Finden innerhalb von Repositorys möglich ist und es somit wiederverwendet werden kann. Metadaten werden durch das Content Packaging (siehe 3.2.1.2) mit einem Asset assoziiert.

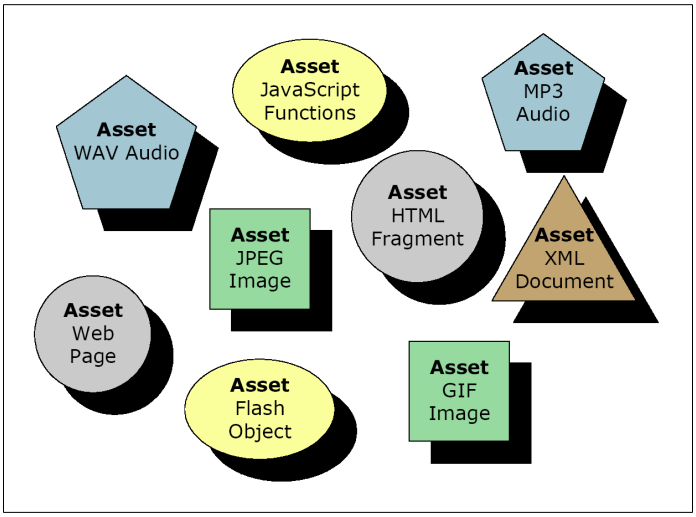


Abb. 3.5: Assets [ADL04b]

Shareable Content Object: Ein SCO ist eine Zusammenstellung aus einem oder mehreren Assets bzw. aus Daten, die nicht extra als Asset definiert sind (siehe Abb. 3.6 auf der nächsten Seite). Der Unterschied zu einem Asset besteht darin, dass ein SCO über die Laufzeitumgebung (Run-Time Environment siehe 3.2.2) mit dem LMS kommuniziert.

Zum Erreichen einer Wiederverwendbarkeit sollte jedes SCO ein vom Gesamtkontext unabhängiges Objekt darstellen und eine abgeschlossene, sinnvolle Lerneinheit bilden. Daher darf es nie direkt auf ein anderes SCO verweisen bzw. auf dessen Daten zugreifen. Eine Größenbeschränkung gibt es nicht, es sollte jedoch darauf geachtet werden, dass innerhalb eines SCOs ein bestimmtes Lernergebnis erreicht werden kann. Der SCORM-Standard macht allerdings keinerlei genauere Angaben, was die Wiederverwendbarkeit des Inhalts selbst betrifft - es liegt am Entwickler den Inhalt unabhängig von seinem Kontext zu machen. Der Grad der Wiederverwendbarkeit ist deshalb direkt vom Entwickler abhängig.

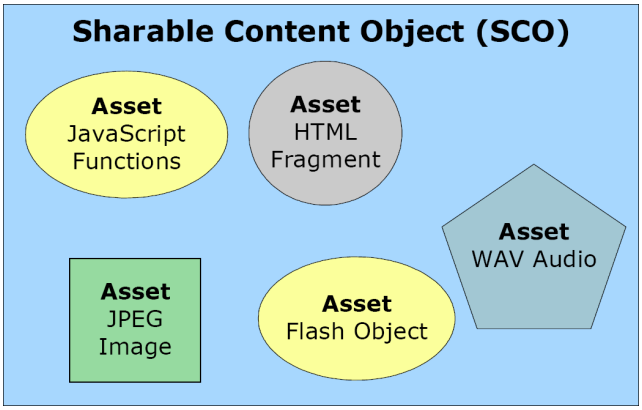


Abb. 3.6: Shareable Content Object [ADL04b]

SCOs können wie Assets auch mit Metadaten beschrieben werden, die Assoziation von Metadaten und SCO erfolgt wiederum durch das Content Package.

Content Organization: Mittels einer *Content Organization* oder Inhaltsstruktur werden einzelne Lern-Ressourcen zu strukturierten und logisch gegliederten Einheiten zusammengefasst; bei dieser hierarchischen Struktur spricht man auch von Content Structure (Lernreihenfolge). Solche Einheiten können ein Kapitel, ein Kurs oder schlicht eine Abfolge von Lerninhalten sein. In SCORM wird die Reihenfolge (Sequenz, siehe Abschnitt 3.3 auf Seite 57) bzw. der Ablauf der Lerneinheiten in der Content Structure definiert; damit ist eine Trennung zwischen Inhalt und Ablauf garantiert und die geforderte Wiederverwendbarkeit erfüllt. Das LMS ist dafür verantwortlich, dass eine korrekte Interpretation der Content Structure erfolgt und somit die SCOs in der richtigen Reihenfolge ablaufen. Weiters kann man mit Content Aggregation Metadaten referenzieren, um die Suche und Wiederverwendbarkeit in beispielsweise Onlinearchiven zu ermöglichen.

3.2.1.2 Content Packaging

Content Packaging wird verwendet, um Inhalte oder digitale Lern-Ressourcen zwischen verschiedenen Lern-Management-Systemen, Entwicklungswerkzeugen und Kursdatenbanken auszutauschen bzw. in beliebigen Repositorys zu speichern. SCORM Content Packaging hält sich strikt an die IMS Content Packaging Specifi-

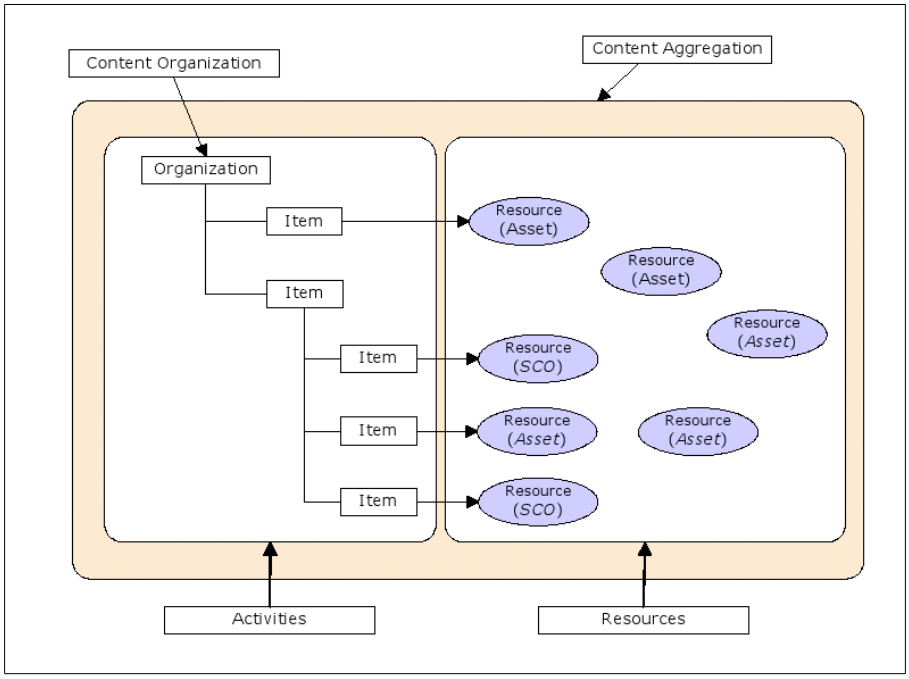


Abb. 3.7: Komponenten des Content Aggregation Model [ADL04b]

kation mit jedoch zusätzlichen Anforderungen und Implementationsrichtlinien für das Packaging digitaler Lern-Ressourcen wie Assets, SCOs und Content Aggregations.

Die so genannte *Content Structure* beschreibt wie aus einzelnen Lern-Ressourcen logisch zusammenhängend aufgebaute Lerneinheiten erstellt werden können. Weiters definiert sie neben der Navigation durch die Lern-Ressourcen auch das weitere Verhalten der Einheit. Die Content Structure umfasst einen großen Teil des Content Aggregation-Verfahren und reicht von sehr kleinen bis zu hoch interaktiven Lern-Ressourcen.

Sie gliedert sich in drei Teile:

Content Hierarchy: Ordnet Lern-Ressourcen logisch innerhalb einer Baumstruktur an (siehe Abb. 3.8 auf der nächsten Seite).

Content Specific Meta-Data: Neben Informationen über den Autor beinhaltet dieser Teil zusätzlich eine Beschreibung des Namens und Zwecks der Lern-Ressource. Diese Beschreibung kann, sofern es sich um eine alleinstehende Ressource handelt, sowohl Kontext-unabhängig als auch Kontext-abhängig

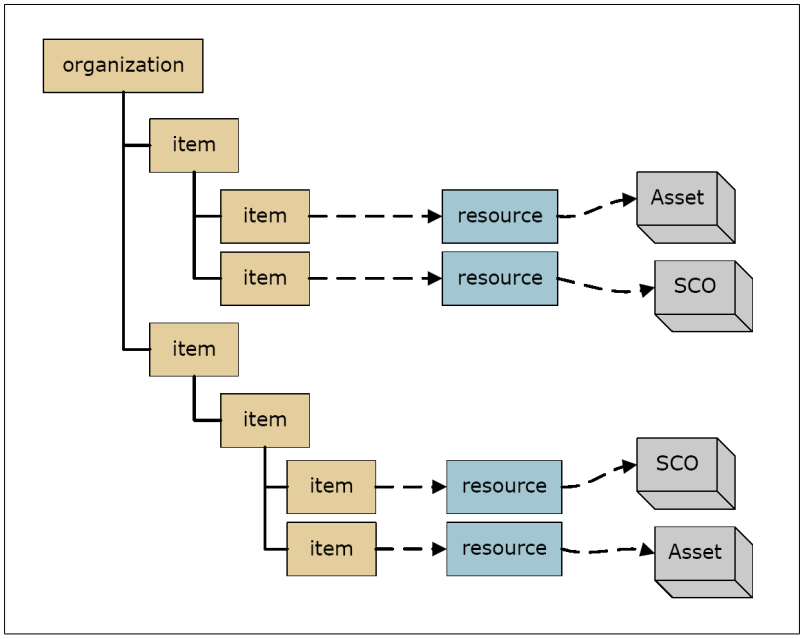


Abb. 3.8: Content Hierarchie [ADL04b]

sein, um zu zeigen, wie und wo diese Lern-Ressource innerhalb einer Sammlung von Ressourcen benutzt werden kann.

Sequencing and Navigation: Dieser Teil ist essentiell für den Ablauf des Lernens. Hier wird bestimmt, *welche* Informationen dem Lernenden *wann* zu präsentieren sind. Dabei muss berücksichtigt werden, welche Lektionen bereits absolviert wurden bzw. welche Auswahlmöglichkeiten dem Benutzer zur Verfügung stehen und welche er bereits getroffen hat. Für die Festlegung der Navigationsinformationen ist derjenige verantwortlich, der die Inhalte zusammenstellt.

Bestandteile eines Content Package (IMS Package Description and Packaging Information Model)

Im wesentlichen besteht ein Content Package aus zwei Teilen: Dem so genannten *Manifest* und den darin referenzierten Dateien (siehe Abb. 3.9 auf der folgenden Seite).

Das Content Package repräsentiert eine selbständige wiederverwertbare Lerneinheit. Für den Transfer, z.B. über das Internet, wird ein spezielles Format, das Package Interchange File (PIF) verwendet. Bei den Dateien kann es sich sowohl um

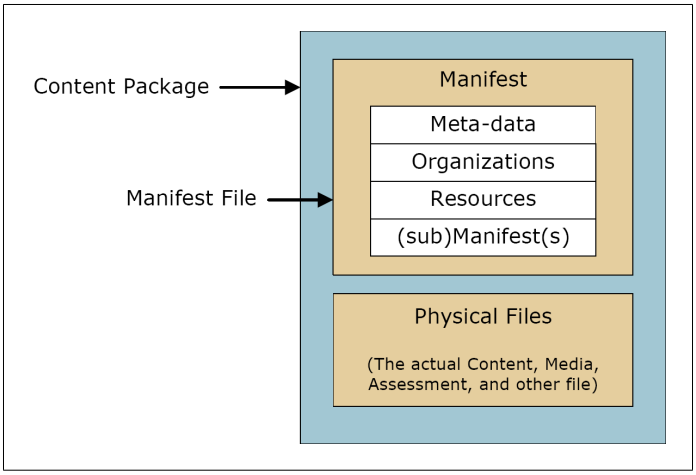


Abb. 3.9: Content Packaging [ADL04b]

physikalisch (lokal) vorhandene Dateien, wie auch um externe Ressourcen handeln, auf die via URI (Uniform Resource Identifier) referenziert wird.

Das Manifest beschreibt das Content Package via XML-Format. Mit dem SCORM Content Packaging XML Binding wird definiert, wie das Informationsmodell in ein XML-Dokument zu konvertieren ist. Listing 3.1 zeigt ein Beispiel einer Manifestdatei (`imsmanifest.xml`), wobei ersichtlich ist, dass die Datei sich in mehrere Abschnitte gliedert:

manifest Für eine XML (Extensible Markup Language)-Datei typische Schema-Beschreibung zur Validierung der Manifest-Datei (Attribute: `identifier`, `version`)

metadata Beschreibt die einzelnen Komponenten eines Content Package auf verschiedenen Ebenen (Attribute: `schema`, `schema version`, `location`, `meta-data`).

organizations Hiermit können beliebige Strukturen integriert und festgelegt werden. Eine Struktur besteht dabei aus dem Wurzelelement `<organization>` und in verschiedenen Tiefen verschachtelte `<item>`-Elemente. (Attribute: `default`, `organization`, `identifier`, `structure`, `title`, `item`)

In diesem Bereich können auch Vorschriften für die Ablaufsteuerung und die Anzeige von Navigationselementen eingefügt werden. Dafür wurden die Elemente `<sequencing>` und `<sequencingCollection>` eingeführt. Mehr dazu in 3.3.

resources Dient zur Referenzierung externer Ressourcen bzw. lokaler Dateien (z.B. Text-, Mediendateien). Es ist möglich, Dateien zu gruppieren bzw. in Beziehung zueinander zu setzen (siehe Abb. 3.10). Diese Form der Kombination von Ressourcen wird allgemein auch als Inhalt (Content) bezeichnet. (Attribute: `ressource`, `identifier`, `type`, `location`, `file`, `dependency`)

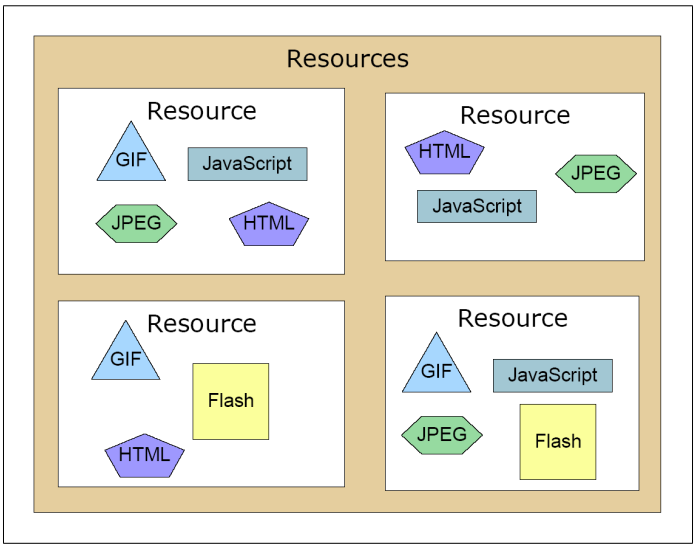


Abb. 3.10: Manifest-Ressourcen [ADL04b]

3.2.1.3 Meta-Data

Metadaten beschreiben die individuellen Eigenschaften der einzelnen Komponenten (Lern-Ressourcen) und sind - in Hinblick auf Wiederverwendbarkeit - unentbehrlich. Erst durch Metadaten ist es möglich, SCOs in Online-Repositorys abzulegen und wiederzufinden.

Im so genannten Metadaten-Anwendungsprofil ist festgelegt, welche Metadaten-Elemente optional und welche obligatorisch sind. Im CAM (aktuelleste Version 1.3.1) gibt es drei verschiedene Arten von Metadaten (siehe Tabelle 3.1); die Profile für Content Organizations, Activities und SCOs sind identisch.

Prinzipiell sind alle Metadaten optional, da auch die `<metadata>`-Elemente in der Manifestdatei optional sind. Werden hingegen Metadaten eingefügt, müssen

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest identifier="example manifest"
  xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"
  xmlns:imsmd="http://www.imsglobal.org/xsd/imsmd_v1p2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_v1p3"
  xmlns:imsss="http://www.imsglobal.org/xsd/imsss"
  xmlns:adlseq="http://www.adlnet.org/xsd/adlseq_v1p3"
  xmlns:adlnav="http://www.adlnet.org/xsd/adlnav_v1p3"
  xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1
    imscp_v1p1.xsd
  http://www.imsglobal.org/xsd/imsmd_v1p2 imsmd_v1p2p2.xsd
  http://www.adlnet.org/xsd/adlcp_v1p3 adlcp_v1p3.xsd
  http://www.imsglobal.org/xsd/imsss imsss_v1p0.xsd
  http://www.adlnet.org/xsd/adlseq_v1p3 adlseq_v1p3.xsd
  http://www.adlnet.org/xsd/adlnav_v1p3 adlnav_v1p3.xsd" version="1.3">

  <metadata>
    <schema>ADL SCORM</schema>
      <schemaversion>CAM 1.3</schemaversion>
  </metadata>

  <organizations>
    <organization identifier="organisation">
      <title>Organization</title>
      <item identifier="sco01" identifierref="res01" isvisible="true">
        <title>Introduction</title>
      </item>
    </organization>
  </organizations>

  <resources>
    <resource identifier="res01" type="webcontent" href="sco_01.html"
      adlcp:scormType="asset">
    </resource>
  </resources>
</manifest>
```

Listing 3.1: Beispiel einer Manifestdatei (`imsmanifest.xml`)

Position innerhalb Manifest	Anwendungsprofil
`<manifest>`	Content Aggregation MD
`<organization>`	Content Organization MD
`<item>`	Activity MD
`<resource> (bei adlcp:scormType=''sco'')`	SCO MD
`<resource> (bei adlcp:scormType=''asset''), <file>`	Asset MD

Tab. 3.1: Metadaten-Anwendungsprofile im CAM 1.3.1

die Anforderungen der Anwendungsprofile eingehalten werden.

Ein SCORM-Metadatensatz besteht gemäß dem LOM-Standard aus neun Abschnitten (*Categories*):

General: Allgemeine Beschreibung über eine Ressource (z.B. Titel, Sprache, Stichwörter)

Lifecycle: Gruppiert die Eigenschaften einer Komponente bzw. die Eigenschaften jener Komponenten, die diese beeinflusst haben. (z.B. Version, Status, Beitrag, Rolle, Datum)

Meta-Metadata: Sammelt Informationen über den Metadatensatz. Beispielsweise kann festgehalten werden, wann die Metadaten von wem eingegeben wurden und in welcher Sprache sie angegeben sind.

Technical: Beschreibt Informationen über technische Anforderungen. (z.B. Format, Größe, Typ, Anforderungen, Installation) und Eigenschaften der Ressource. Zum Beispiel kann hier eine URL angegeben werden, unter der die neueste Version des Inhalts heruntergeladen werden kann.

Educational: Beschreibt die pädagogischen Eigenschaften der Komponente. Dies kann von Lehrern, Autoren, Managern und Lernenden genutzt werden. (z.B. Typ, Level, Kontext, typische Altersgruppe, Schwierigkeitsstufe, Zeitaufwand, Beschreibung, Sprache)

Rights: Beschreibt Rechte für geistiges Eigentum und Bedingungen zur Nutzung der Ressource. (z.B. Kosten, Copyright, Beschreibung)

Relation: Legt Beziehungen zwischen dieser und anderen Ressourcen fest. (z.B. Art, Ressource, Identifikation, Beschreibung, Katalogeintrag)

Annotation: Anmerkungen des Autors (z.B. Person, Datum, Beschreibung)

Classification: Bestimmt, wo die Ressource eingesetzt werden kann. (z.B. Zweck, Quelle, Gruppe, Eintrag, Beschreibung, Stichwort)

Alle genannten Kategorien entsprechen XML-Elementen (IMS Learning Resource Meta-Data XML Binding Specification), die alle unter einem gemeinsamen Wurzelelement, gemäß LOM , angeordnet sind (siehe Listing 3.2). Jedes der neun Elemente umfasst wiederum bis zu sieben Datenelemente.

```
<lom xmlns="http://ltsc.ieee.org/xsd/LOMv1p0">
  <general>
    <title>
      <string xml:lang="de">Technische Univerität Graz</string>
      <string xml:lang="en">Graz University of Technology</string>
    </title>
    <language>de</language>
  </general>
  <technical>
    <location type="URI">
      http://www.tugraz.at
    </location>
  </technical>
</lom>
```

Listing 3.2: Beispiel für Metadaten

Die Kategorien beinhalten folgende Datenelemente (*Simple and Aggregate Data Elements*):

Name: Name des Datenelements

Explanation: Definition des Datenelements

Size: Anzahl der erlaubten Werte

Order: Zeigt an, ob die Reihenfolge der Werte signifikant ist

Example: ein bildhaftes Beispiel

Zusätzlich werden einzelne Datenelemente (*Simple Data Elements*) mit folgenden Merkmalen beschrieben:

Value Space: Menge erlaubter Werte, normalerweise in Form von Vokabeln oder einer Referenz zu einem anderen Standard

Datatype: Es gibt folgende Werte-Typen: `LangString`, `DateTime`, `Duration`, `Vocabulary`, `CharacterString` oder `Undefined`.

Manche Datenelemente beinhalten eine Werteliste (*List of Values*) anstatt nur eines Wertes. Diese Liste hat einen der beiden folgenden Charakter:

ordered: Die Reihenfolge der Werte ist signifikant.

unordered: Die Reihenfolge der Werte ist nicht von Bedeutung.

Grundsätzlich müssen nicht immer alle Datenelemente angegeben werden. Je nach Komponente, sind einige Metadaten zwingend erforderlich, optional oder nicht verwendbar.

3.2.2 Run-Time Environment

Das Run-Time Environment (RTE) beinhaltet Richtlinien für das Erstellen einer Web-basierenden Umgebung und beschreibt sowohl die Kommunikation als auch das Tracking der Inhalte.

Bedingt durch die bereits geforderte Wiederverwendbarkeit von Lern-Ressourcen ist eine standardisierte Ausführung bzw. eine standardisierte Kommunikation mit dem LMS notwendig.

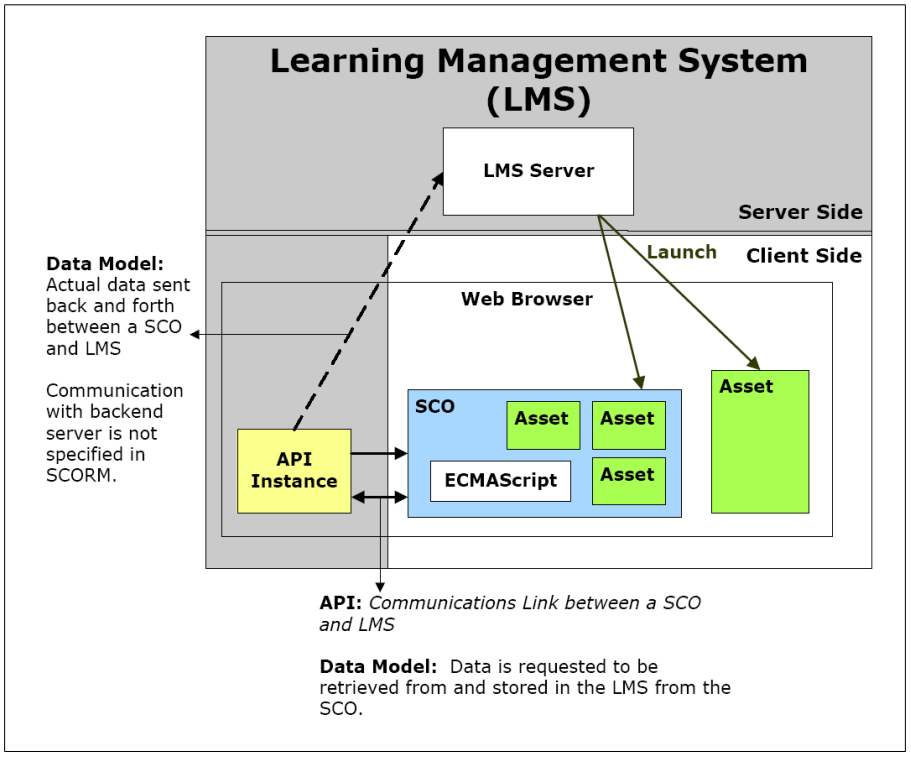


Abb. 3.11: SCORM Laufzeitumgebung [ADL04c]

Die Laufzeitumgebung (siehe Abb. 3.11 auf der vorhergehenden Seite) zum Ausführen der Lern-Ressourcen besteht aus drei Bestandteilen:

Launch Mechanism: Definiert den Startmechanismus mit dem Web-basierte Lern-Ressourcen (Assets oder SCOs) innerhalb des LMS gestartet und via Browser zur Anzeige gebracht werden. Sofern es sich bei den Ressourcen um SCOs handelt, wird auch der Kommunikationsaufbau beschrieben und Prozeduren festgelegt.

API: Das API (Application Programming Interface) ermöglicht die Kommunikation zwischen dem SCO und dem LMS. Dabei informiert es das LMS über den Status der Lern-Ressource und transportiert Daten zwischen LMS und SCO.

Data Model: Definiert die Struktur der zu übertragenden Daten und beschreibt das Datenformat.

3.2.2.1 Launch Mechanism

Um einen Lerninhalt zu starten, muss das LMS in der Lage sein, die nächste zu startende Lernaktivität festzustellen. Mithilfe des Content Package findet das LMS, verursacht durch ein Navigationsereignis, jenen Inhalt, der mit dieser neuen Aktivität verknüpft ist. Im Content Package sind Lerninhalte mit Ressourcen verbunden, die wiederum mittels URLs auf Dateien verweisen. Danach wird der aktuelle Inhalt mit dem neuen ersetzt, indem der alte Inhalt beendet und der neue gestartet wird. Die Reihenfolge und die Navigation wird dabei dem LMS überlassen ist. Über das Layout selbst gibt es keinerlei Vorschriften - die Gestaltung obliegt dem LMS.

Beim Laden der verschiedenen Inhaltsobjekte wird unterschiedlich vorgegangen:

Laden von Assets

Assets (siehe Abschnitt 3.2.1.1 auf Seite 40) stellen nur passiven Inhalt dar, daher fordert der Standard lediglich, dass das LMS diese über das Hypertext Transfer Protocol startet. Es erfolgt keine Kommunikation via API und dem Datenmodell mit dem LMS.

Laden von SCOs

Der Standard verlangt, dass SCOs, der aktive Inhalt in SCORM, vom LMS gestartet werden; dadurch ist das SCO verantwortlich, die API des LMS zu finden. Bei der Ausführung und Lokalisation des Application Programming Interface bestehen allerdings einige Einschränkungen:

- Es darf nur jeweils ein SCO für einen Lernenden aktiv sein.
- SCOs können keine anderen SCOs ausführen.
- Das LMS muss entweder das SCO in einem abhängigen Browserfenster (Popup) starten oder es muss vom LMS in einem Browser-Fenster gestartet werden, das ein Kind-Fenster oder Kind-Frame des LMS-Fensters ist, um Zugriff auf dessen Document Object Model[3] zu haben.

Dem gesamten Launch-Prozess liegt ein Zeitmodell zu Grunde und definiert Begriffe, die für die Verwaltung des RTE (Run-Time Environment) eines SCOs von Bedeutung sind:

Learner Attempt: Bezeichnet den Versuch eines Lernenden, eine Activity zu bearbeiten, wobei er sich über eine oder mehrere Learner Sessions erstecken kann. Ein Learner Attempt beginnt mir der Bereitstellung einer Activity und endet, wenn die Bearbeitung des SCOs abgeschlossen ist.

Learner Session: Ist eine zusammenhängende Zeitspanne, in welcher der Lernende auf ein Content Object zugreift.

Communication Session: Bezeichnet jene Zeitspanne, in welcher eine aktive Verbindung zwischen einem SCO und der API besteht.

Login Session: Umfasst jenen Zeitraum, in dem der Lernenden im LMS eingeloggt ist.

[3] Im DOM (Document Object Model) hat die API den Namen `API_1484_11` und kann somit eindeutig gefunden werden.

3.2.2.2 API

Das API (Application Programming Interface) ermöglicht einen standardisierten Datentransfer zwischen SCOs und dem LMS (siehe Abb. 3.11 auf Seite 50), indem es verschiedene Funktionen als Schnittstelle zur Verfügung stellt. Die Kommunikation erfolgt ausschließlich von einem SCO ausgehend - für ein LMS besteht keine Möglichkeit, Funktionen des SCOs auszuführen. Dabei wird vom LMS eine API-Instanz zur Verfügung gestellt, wodurch Entwickler von Lerneinheiten nur die standardisierten Schnittstellen benutzen brauchen und sich nicht mit Details der Kommunikation beschäftigen müssen.

Die Implementierung der API innerhalb eines LMS ist von einer Programmiersprache unabhängig, ein Zugriff durch ein SCO auf eine API-Instanz ist jedoch genau definiert und erfolgt über ECMAscript (JavaScript)[4].

Der API-Adapter bietet drei Gruppen von Funktionen, die wiederum insgesamt acht API-Funktionen implementieren und den Großteil des SCORM Datenmodells unterstützen:

Session-Methoden: Hiermit kann der Zustand (siehe Abb. 3.12 auf der nächsten Seite) eines SCOs festgelegt werden.

Mögliche Funktionen sind:

- `Initialize('')`: Dient zur Initialisierung der Kommunikation zwischen einem SCO und LMS. Der Parameter ist der leere String ('').
- `Terminate('')`: Die Kommunikation zwischen einem aktiven SCO und dem LMS wird beendet und wird vom SCO selbst aufgerufen. Darauffolgend speichert das LMS die vom SCO empfangenen Daten – meist in einer Datenbank – nach einem implizitem Aufruf der Methode `Commit()`.

Für ein SCO ist es nur erforderlich, diese beiden Methoden aufzurufen, alle folgenden sind optional.

Datentransfer-Methoden: Hierbei handelt es sich um Funktionen für den Datentransfer von und zum LMS.

Mögliche Funktionen sind:

[4] Der zugrunde liegende Standard ist der *IEEE P1484.11.2 - Standard for Learning Technology - ECMAscript Application Programming Interface for Content to Runtime Services Communication*.

- `GetValue(data model element)`: Ein SCO kann mit Hilfe dieser Funktion Daten vom LMS anfordern; die Daten sind Parameter des Datenmodells oder spezieller Werte verschiedener Kategorien und Elemente.
- `SetValue(data model element, value)`: Mit diesem Befehl kann ein SCO Daten an das LMS senden; optional kann das API mit einem Cache für diese Daten implementiert werden.
- `Commit('' '')`: Dieser Befehl sendet eventuelle in der API-Instanz zwischengespeicherten Daten (Cache) an das LMS. Ein Hauptargument für diese Funktion besteht in dem Umgehen des zeitraubenden Zwischenspeicherns bei der Kommunikation zwischen Client (Browser) und LMS. Parameter ist hier der leere String (’’ ’’).

Support-Methoden: Fehler können mit folgenden Methoden behandelt werden.

Mögliche Funktionen sind:

- `GetLastError()`: Diese Funktion überprüft, ob ein Funktionsaufruf erfolgreich war. Sofern dies nicht der Fall ist, wird ein spezieller Fehlercode ausgeführt. Nach jedem Funktionsaufruf, ausgenommen `LMSgetErrorString` und `LMSGetDiagnostic` wird der Fehlercode zurückgesetzt.
- `GetErrorString(error code)`: Diese Methode liefert eine textliche Beschreibung des übergebenen Fehlercodes.
- `GetDiagnostic(value)`: Hiermit ist es möglich, herstellerspezifische Fehlercodes zu integrieren und dem SCO zur Verfügung zu sellen.

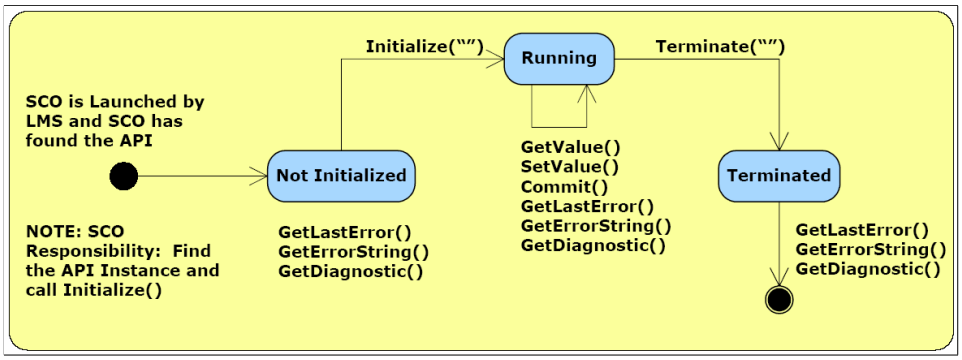


Abb. 3.12: API-Adapter Zustandsdiagramm [ADL04c]

```
<html>
  <head>
    <script language="JavaScript">
      function doInitialize()
      {
        var api = findAPI();
        api.Initialize("");
      }
      function doTerminate()
      {
        var api = findAPI();
        api.Terminate("");
      }
      function findAPI()
      {
        ...
        return APIHandle;
      }
    </script>
  </head>
  <body onload="doInitialize()" onunload="doTerminate()">
    Inhalt
  </body>
</html>
```

Listing 3.3: Beispiel für API-Funktionsaufrufe - Initialisierung und Beendigung der Kommunikation zwischen LMS und SCO

Communication Session State Model

Ein SCO bzw. dessen API-Instanz kann drei Zustände annehmen (siehe Abb. 3.12 auf der vorhergehenden Seite):

Not Initialized: Der Zustand „not initialized“ ist zwischen dem Start des SCOs und dem Aufruf der Methode `Initialize()` aktiv. Während dieses Zustandes muss das SCO die API-Instanz des LMS suchen. Folgende Methoden dürfen vom SCO aufgerufen weden:

- `Initialize()`
- `GetLastError()`
- `GetErrorString()`
- `GetDiagnostic()`

Running: Sobald die Methode `Initialize()` aufgerufen wurde, gilt der Zustand „running“ - solange bis dieser durch die API-Methode `Terminate()` durch das SCO beendet wird. Folgende Methoden stehen zur Auswahl:

- `Terminate()`
- `GetValue()`
- `SetValue()`
- `Commit()`
- `GetLastError()`
- `GetErrorString()`
- `GetDiagnostic()`

Terminated: Nach Aufruf von `Terminate()` gilt der Zustand „terminated“. Mittels folgender Methoden können Informationen eingeholt werden:

- `GetLastError()`
- `GetErrorString()`
- `GetDiagnostic()`

3.2.2.3 Data Model

Das SCORM Datenmodell stellt einen standardisierten Satz[5] von Elementen zur Verfügung, mit denen zur Laufzeit über die API-Instanz Informationen zwischen aktivem Inhalt (SCOs) und LMS ausgetauscht werden können. Mit der Benutzung eines einheitlichen Datenmodells wird eine übereinstimmende Informationsverarbeitung – Tracking von SCOs – in verschiedenen LMS sichergestellt.

Alle Elemente des Datenmodells müssen in der aktuellsten SCORM-Version von einem LMS unterstützt werden; für SCOs sind die Elemente hingegen optional. Die Kennzeichnung der zum Datenmodell gehörenden Elemente erfolgt mit dem Präfix `cmi`, danach folgt der Elementname dem hierarchischen Aufbau des Modells folgend. Jedem Datenelement ist ein Datentyp zugeordnet. Es folgt nun eine überblicksmäßige Auflistung der verschiedenen Typen. Eine genaue Beschreibung der Datenelemente ist aus [ADL04c] zu entnehmen.

[5] Als Grundlage dient der *IEEE 1484.11.1 - Draft Standard for Data Model for Content Object Communication*.

```
1 SetValue(cmi.score.scales,5)
```

Listing 3.4: Beispiel für API-Aufruf eines SCOs mit der API-Funiktion `SetValue()`, dem Parameter `cmi.score.scaled` und dem Score `5`

Datenmodelle können in verschiedene Kategorien unterteilt werden:

Core: Allgemeine Informationen über den Lernenden und über das Unterrichtsmaterial

Suspend Data: Vom LMS gespeicherte SCO-Daten einer früheren Benutzung

Launch Data: Informationen, die für den Start eines SCOs benötigt werden

Comments: Freier Text als Feedback bzw. zum Informationsaustausch zwischen SCO und LMS

Objectives: Informationen über die Zielsetzung eines SCOs

Student Data: Informationen zur Personalisierung des SCOs - abhängig von der vom Lernenden erbrachten Leistung

Interactions: Registrierte Eingaben des Lernenden

Student Preference: Optionen, die der Lernende selbst auswählen kann

Version: Repräsentiert die Version des Datenmodells und sollte den Wert „1.0“ enthalten

3.3 Sequencing und Navigation

Im folgenden Kapitel werden grundlegende Konzepte und Methoden der SCORM 2004 (Version 1.3.1) Ablaufsteuerung und Navigation erläutert. Sequencing wird dabei detaillierter betrachtet, Navigation nur kurz umrissen. Abschließend erfolgt ein Vergleich zum Sequencing and Navigation in der Version 1.2.

3.3.1 Einführung

Vorrangiges Ziel von SCORM ist es, Lernangebote so zu gestalten, dass diese auch wiederverwendet werden können. In den meisten Fällen sind jedoch Struktur und Ablaufsteuerung untrennbar mit dem Inhalt verknüpft. Mit Sequencing wurde nun eine Möglichkeit geschaffen, die Ablaufsteuerung vom Inhalt zu trennen und so eine Wiederverwendbarkeit von Lernmaterialen und eine unabhängige Gestaltung der Ablaufsteuerung zu schaffen.

Währernd in der Version 1.2 noch eine relativ einfache Form der Ablaufsteuerung implementiert und im CAM (Content Aggregation Model) Buch enthalten war, wurde ab SCORM 2004 das Referenzmodell um ein viertes, eigenständiges Buch *Sequencing and Navigation* erweitert; es basiert teilweise auf der *IMS Simple Sequencing (SS)* Spezifikation. Das Buch umfasst folgenden Themengebiete:

Konzepte und Terminologie: Beinhaltet eine Beschreibung und Definition grundlegender Begriffe

Sequencing Definition Model: Beschreibt die Sequencing-Steuerungsregeln, die auf Lernaktivitäten angewandt werden können.

Sequencing Behaviour Model: Beschreibt, welche Prozesse und welches Verhalten SCORM-konforme LMS umsetzen müssen, um das Sequencing zu steuern.

Navigation Data Model: Beschreibt, wie Navigation und Benuteroberfläche vom Content-Entwickler beeinflußt werden können.

Sequencing bezieht sich dabei stets nur auf die Ablaufsteuerung zwischen einzelnen Activities, die Navigation innerhalb eines Content Objects einer Activity wird vom LMS nicht gesteuert und damit auch nicht erfasst. Im folgenden werden wichtige Konzepte des Sequencing vorgestellt.

3.3.1.1 Activity

Eine *Activity* oder (Lern-)Aktion ist etwas, was ein Lernender, während er durch das Lernabgebot fortschreitet, „tut“. Eine Aktion entspricht entweder einem Blatt im so genannten Activity Tree (siehe 3.3.1.2) verbunden mit einer Lern-Ressource

oder besteht aus weiteren Sub-Aktionen, die wiederum aus weiteren Activities aufgebaut sein können.

Eine solche Verschachtelung von Aktionen wird Cluster genannt. Ein Cluster umfasst die Eltern-Activity und ihre direkten Kind-Activities, deren Kinder zählen jedoch nicht mehr dazu. Kinder eines Clusters sind also entweder Blatt-Activities und verweisen auf eine Lernressource oder sind selbst wieder ein Cluster (siehe Abb. 3.13).

Welche Activity dem Lernenden vom LMS als nächste angezeigt wird, wird zur Laufzeit des Systems bestimmt und hängt vom Fortschritt des Lernenden in vorherigen Aktivitäten, der Lernabsicht und den vorgegebenen Ablaufsteuerungsinformationen (siehe Sequencing Rules) ab.

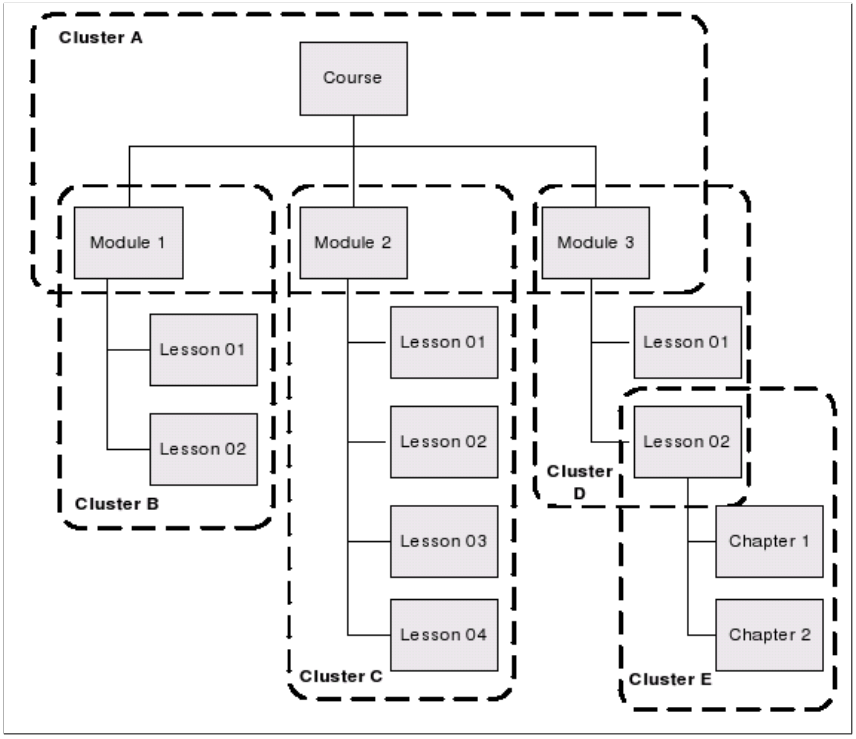


Abb. 3.13: Schema eines Activity Trees mit Clustern [ADL04d]

3.3.1.2 Activity Tree

Ein *Activity Tree* oder Aktionsbaum wird direkt aus der Inhaltsstruktur (Content Organization, siehe 3.2.1.1) abgeleitet und beschreibt die Struktur der Lernaktivitäten. Er repräsentiert quasi eine Instanz von hierarchisch aufgebauten Lernaktivitäten und zugehörigen Ablaufsteuerungsinformationen.

Aufbau

Ausgehend vom Manifest (siehe 3.2.1.2) bildet das `<organization>`-Element die Wurzel-Activity, jedes untergeordnete `<item>`-Element entspricht einer Aktion; diese enthalten auch die Ablaufsteuerungsregeln, welche in `<sequencing>`-Elementen oder `<sequencingCollection>`-Elementen gekapselt sind (siehe Listing 3.5). Es existiert für jeden Lehrer und für jeden Kurs ein Activity Tree.

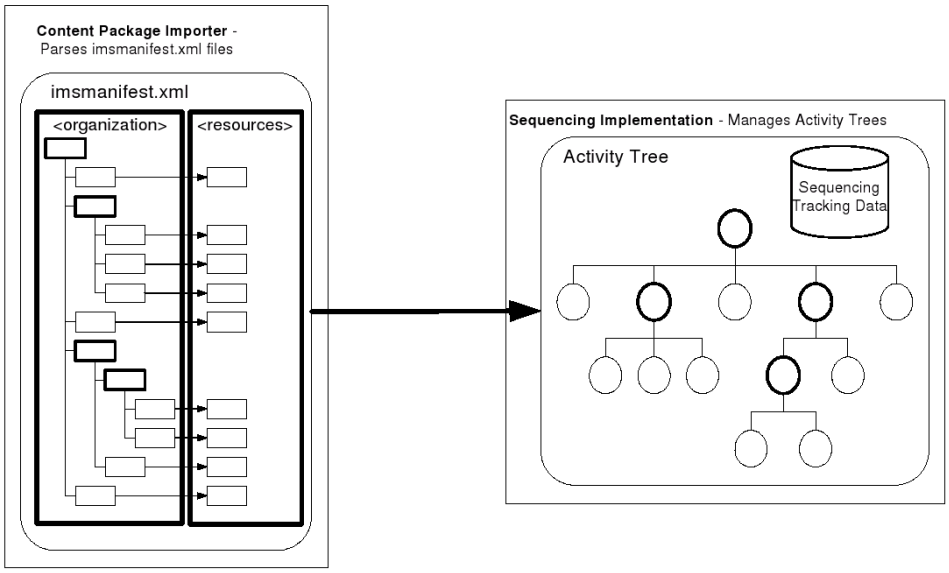


Abb. 3.14: Ableitung des Activity Trees aus der Content Organization [ADL04d]

3.3.1.3 Attempts

Ein Versuch oder *Attempt* tritt zeitgleich mit dem Bearbeiten einer Activity auf und versucht diese zu lösen. Je nach Position der Aktion im Activity Tree, werden alle auf dem Weg zur Wurzelaktion des Activity Trees befindlichen Versuche gestartet. Im Baum darf nur eine Blatt-Activity pro Lernendem versucht werden; durch einen Attempt können ein oder mehrere Lernziele erreicht werden. Weiters besteht noch die Möglichkeit, eine Activity zu unterbrechen, wodurch es jedoch zu unvollständigen Activities kommen kann.

3.3.1.4 Learning Objectives

Unter *Learning Objectives* oder Lernzielen versteht man Variablen, mit welchen Statuswerte wie beispielsweise „bestanden, nicht bestanden, Ergebniswerte“ gespeichert werden; im Prinzip kann damit der Lernerfolg eines Lernenden festgehalten werden. Aktionen können mit einem oder mehreren Lernzielen verbunden werden, die Objectives können wiederum von mehreren Activities gemeinsam benutzt werden.

Prinzipiell unterscheidet man zwischen lokalen und globalen Objectives. Lokale sind an eine Activity gebunden und können auch nur von dieser genutzt werden; globale hingegen können von allen Activities genutzt werden und sind nicht ausschließlich an eine Aktion gebunden. Mit dem Element `<objectives>` werden Objectives den Activities zugeordnet.

3.3.2 Sequencing Definition Model

Das SCORM Sequencing Definition Model stellt eine Reihe von Elementen zur Verfügung, mit denen Entwickler von Lerneinheiten Sequencing-Strategien festlegen können. Ein Entwickler hat damit die Möglichkeit zu bestimmen, wann einem Lernenden welche Inhalte präsentiert werden; gleichzeitig bekommt das LMS Informationen zur Steuerung des Ablaufs, der Auswahl und Bereitstellung der Aktionen.

Das CAM Buch (siehe 3.2.1) beschreibt, wie Sequencing-Elemente innerhalb der Manifest-Datei eingefügt werden. Mit dem `<sequencing>`-Element können

```
<item identifier="Module1">
  <item identifier="EXAM1">
    <title>Module 1 -- Exam</title>
      <item identifier="QUESTION1" isvisible="false"
            identifierref="RESOURCE_QUESTION1">
        <title>Question 1</title>
      </item>
      <item identifier="QUESTION2" isvisible="false"
            identifierref="RESOURCE_QUESTION2">
        <title>Question 2</title>
      </item>
      <item identifier="QUESTION3" isvisible="false"
            identifierref="RESOURCE_QUESTION3">
        <title>Question 3</title>
      </item>
      <imsss:sequencing>
        <imsss:controlMode choice="false" choiceExit="false"
                           flow="true" forwardOnly="true"/>
        <imsss:rollupRules>
          <imsss:rollupRule childActivitySet="all">
            <imsss:rollupConditions>
              <imsss:rollupCondition condition="attempted"/>
            </imsss:rollupConditions>
            <imsss:rollupAction action="completed"/>
          </imsss:rollupRule>
        </imsss:rollupRules>
        <imsss:objectives>
          <imsss:primaryObjective satisfiedByMeasure="true">
           <imsss:minNormalizedMeasure>0.6</imsss:minNormalizedMeasure>
          </imsss:primaryObjective>
        </imsss:objectives>
      </imsss:sequencing>
  </item>
  <imsss:sequencing>
    <imsss:controlMode flow="true"/>
      <imsss:sequencingRules>
        <imsss:exitConditionRule>
          <imsss:ruleConditions>
            <imsss:ruleCondition condition="completed"/>
          </imsss:ruleConditions>
          <imsss:ruleAction action="exit"/>
        </imsss:exitConditionRule>
      </imsss:sequencingRules>
    <adlseq:constrainedChoiceConsiderations constrainChoice="true"/>
  </imsss:sequencing>
</item>
```

Listing 3.5: Beispiel für Sequencing-Informationen in der Manifest-Datei

grundsätzlich jedem `<item>`-Element und `<organization>`-Element Sequencing-Informationen zugewiesen werden (siehe Listing 3.5). Jedes Element ist mit einem Namespace-Präfix gekennzeichnet: `imsss` sofern aus der IMS-Spezifikation stammend bzw. `adlseq` als zusätzliche von ADL eingeführte Elemente. Weiters werden jedem Element Default-Werte zugewiesen, sofern kein anderer Wert explizit gesetzt wurde.

Es folgt nun eine kurze Beschreibung von Steuerungsmöglichkeiten innerhalb der Manifest-Datei:

3.3.2.1 Sequencing Control Modes

Sequencing Control Modes dienen zur Spezifizierung von Elementen in Navigationsmenüs angewandt auf Cluster. Ein Entwickler kann den Weg durch den Aktionsbaum festlegen und dabei bestimmen, welche Aktionen direkt im Menü ausgewählt werden dürfen. Die Elemente haben einen binären Wertebereich, sie besitzen als nur die Werte `true` oder `false`.

Sequencing Control Choice: Hat als Default-Wert `true`, der Lernende kann jede Activity innerhalb eines Clusters in beliebiger Reihenfolge und ohne Beschränkung auswählen, sofern dies nicht durch andere Steuerungsmodule verhindert wird (siehe Abbildung 3.15).

Sequencing Control Choice Exit: Erlaubt die Terminierung dieser Activity, falls eine andere Activity ausgewählt wird. (Default-Wert: `true`)

Sequencing Control Flow: Erlaubt die Navigation durch die Kinderaktionen eines Clusters mittels „Vor-“ und „Zurück-“Buttons. Falls das Attribut den Wert `true` hat, muss das LMS dem Lernenden Möglichkeiten zur Verfügung stellen, um vorangegangene und nachfolgende Activities auswählen zu können (siehe Abbildung 3.15). (Default-Wert: `false`)

Sequencing Control Forward Only: Falls `true`, kann der Lernende nur vorwärts durch die Activities des Clusters navigieren; es ist nicht erlaubt, zu vorhergehenden Activities zurückzuwechseln. (Default-Wert: `false`)

Use Current Attempt Objective Information: Gibt an, ob während verschiedener Sequencingprozesse die Objective Progress Information (siehe Unterab-

schnitt 3.3.3) des aktuellen oder des vorangegangenen Zugriffs auf eine Activity verwendet wird. (Default-Wert: `true`)

Use Current Attempt Progess Information: Gibt an, ob während verschiedener Sequencing-Prozesse die Attempt Progress Information (siehe Unterabschnitt 3.3.3) des aktuellen oder des vorangegangenen Zugriffs auf eine Activity verwendet wird. (Default-Wert: `true`)

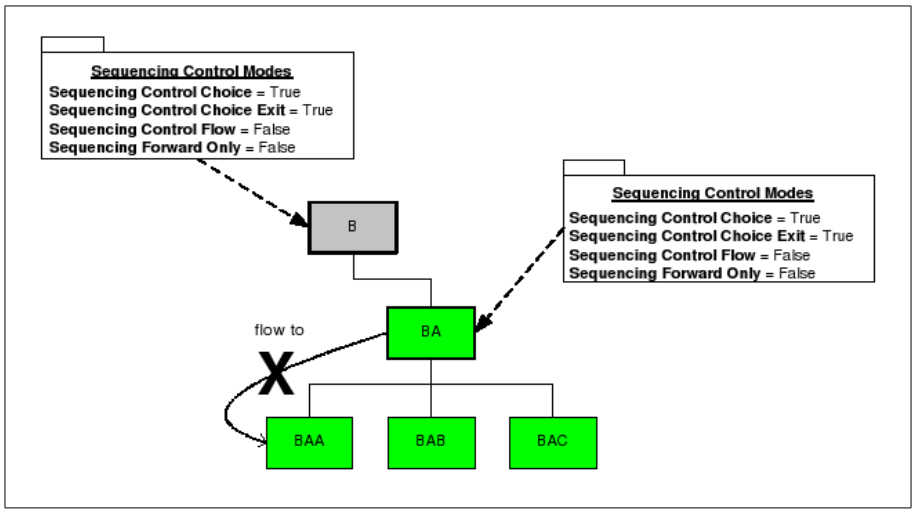


Abb. 3.15: Cluster Activity mit Choice [ADL04d]

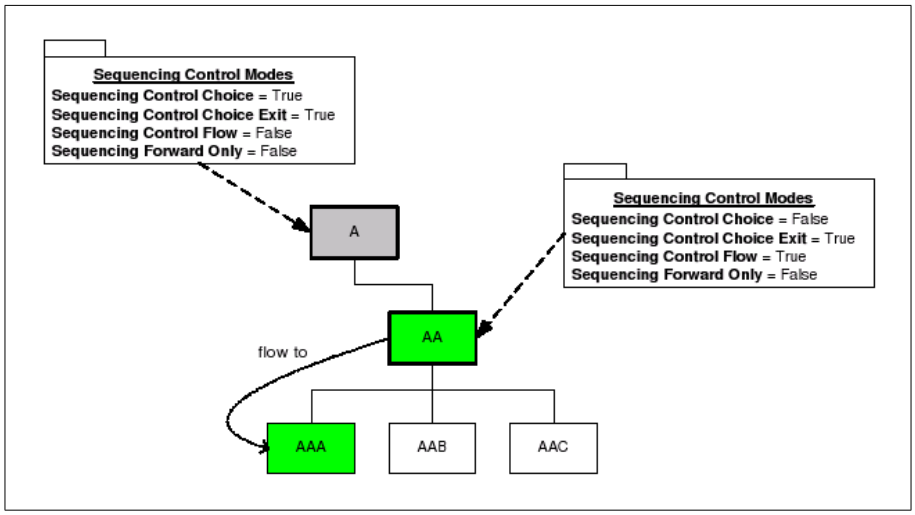


Abb. 3.16: Cluster Activity mit Flow [ADL04d]

Die Sequencing Control Modes können miteinander kombiniert und für jede Activity innerhalb des Activity Trees angewandt werden. Von einem guten LMS wird verlangt, dass es dem Lernenden verbotene Navigation Requests von vornherein verbietet.

3.3.2.2 Constrain Choice Controls

Activities, deren Parent-Activity ein Seqencing Control Choice mit dem Wert `true` besitzt, können theoretisch im ganzen Baum ausgewählt werden. Nachdem diese Flexibilität nicht immer sinnvoll ist, vor allem bei einer abschließenden Testeinheit, wurden die Constrain Choice Controls eingeführt:

Constrain Choice: Beschränkt die auszuwählenden Activities nur auf jene, die sich direkt vor oder nach der aktuellen Activity befinden. Dadurch soll verhindert werden, dass der Lernende beliebig tief im Activity Tree navigiert und so viele Aktionen auslässt. (Default-Wert: `false`)

Prevent Activation: Activities eines Clusters sind erst dann auswählbar, wenn das Wurzel-Element des Clusters aktiv ist; also beispielsweise wenn es über Vorwärts-Navigieren (Flow Navigation Requests) erreicht wurde. (Default-Wert: `false`)

3.3.2.3 Sequencing Rules

Die IMS-SS-Spezifikation bietet ein Rollen-basiertes Modell, damit ist es möglich, für Aktionen eine oder mehrere Sequencing Rules festzulegen. Diese Regeln bestimmen, wann zur Laufzeit etwa Activities übersprungen, im Navigationsmenü angezeigt oder verlassen werden. In der Manifestdatei sind die Regeln durch das entsprechende Element `<sequencingRules>` (siehe Listing 3.5) implementiert.

Jede Regel ist aus einer oder mehreren Bedingungen und einer bestimmten Aktion (`<ruleAction>`) aufgebaut. Bedingungen werden mittels Tracking-Informationen (siehe Unterabschnitt 3.3.3) ausgewertet und der gesamte Bedingungsteil einer Regel wird auf den Wert `true` bzw. `false` reduziert. Sofern `true`, wird die angegeben Aktion ausgeführt. Entsprechend dem Zeitpunkt der Prüfung gibt es drei Arten von Regeln:

- `<preConditionRule>` wird ausgeführt, wenn die nächste bereitzustellende Activity ermittelt wird.
- `<postConditionRule>` wird bei Beendigung des Zugriffs auf die Activity ausgeführ.
- `<exitConditionRule>` wird ausgewertet, wenn der Zugriff auf eine untergeordnete Activity beendet wird.

3.3.2.4 Limit Conditions

Für einen Content-Entwickler besteht die Möglichkeit, Bedingungen zu definieren unter welchen eine Aktion nicht bereitgestellt werden darf. Die Die IMS-SS-Spezifikation enthält zeitbasierende Bedingungen und eine Beschränkung der Anzahl an Zugriffen auf eine Activity.

Das Element `<limitConditions>` verfügt über folgende Attribute:

attemptLimit: Gibt an, wie oft eine Aktion bereitgestellt werden darf.

attemptAbsoluteDurationLimit: Muss als einzige zeitbasierende Bedingung von SCORM-konformen LMS unterstützt werden. Es wird damit die maximale Zeitdauer, die ein Lernender beim Versuchen einer Activity benötigen darf, definiert.

3.3.2.5 Rollup Rules

Unter *Rollup* versteht man ein Weiterreichen von Lernergebnissen – meist Tracking-Information – in unteren Teilen des Activity Trees an weiter oben liegende Cluster-Aktivitäten. Cluster-Activities sind nämlich nicht mit einem Content Object (Inhaltsobjekt) verbunden, daher ist es für solche Activities nicht möglich, die Objective Progress Information und Attempt Progress Information direkt zu setzten.

Jeder Cluster-Activity können eine oder mehrere Rollup Rules zugeordnet werden, die während des *Overall Rollup Process* (siehe Unterabschnitt 3.3.4) ausgewertet werden. Dieser Prozess wird bei Beendigung einer Aktion oder Statusänderung eines globalen Lernziels ausgelöst.

Im Manifest-File werden Rollup Rules innerhalb des Elements `<rollupRules>` definiert. Jede Regel besteht dabei aus den zu berücksichtigenden Kind-Aktionen, einer oder meherer Bedingungen und der Aktion selbst.

Das Element `<rollupRules>` umfasst alle Rollup Rules für eine Activity und verfügt über drei Attribute, die Rollup Controls. Damit besteht für einen Entwickler die Möglichkeit festzulegen, ob und wie eine Activity für das Rollup ihrer Cluster-Activity einbezogen wird. Weitere Elemente innerhalb des Elements `<rollupRules>` sind:

rollupRule definiert eine einzige Rollup Rule und gibt an, wie die Eigenschaften der Kind-Aktionen zu berücksichtigen sind.

rollupConditions ist das Container-Element für die Bedingungen einer Rollup Rule - `<rollupCondition>` definiert eine einzige Bedingung. An dieser Stelle ist es möglich, zu bestimmen, wie Bedingungen miteinander kombiniert werden können. Jede Bedingung kann mit einem Operator versehen werden.

rollupAction enthält jene auszuführende Aktion, wenn das Element `<rollupCondition>` den Wert `true` besitzt.

3.3.2.6 Rollup Consideration Controls

Prinzipiell sind alle Kind-Aktionen in das Rollup der Cluster-Aktion einbezogen, außer die Aktion wird nicht registriert oder durch Rollup Controls vom Rollup ausgeschlossen. In diesem Fall kommt es beim Auswerten der Tracking-Informationen zum Status „unknown“. Um dies zu verhindern wurden von ADL die Rollup Consideration Controls eingeführt - übersprungene, ausgeblendete oder unterbrochene Activities können somit ausgeschlossen werden. Mit dem Element `<rollupConsiderations>` können die Kontrollelemente der jeweiligen Aktion zugeordnet werden. Es kann mir mehreren Attributen versehen werden, mit denen bestimmt werden kann, wann Tracking-Informationen in das Rollup einbezogen werden.

3.3.2.7 Selection Controls

Content-Entwickler besitzen die Möglichkeit, Sequencing Information zu definieren, die bestimmt, wann bestimme Activities ausgewählt werden bzw. ob die Anzahl der auswählbaren Activities eingeschränkt werden soll.

Es werden drei Elemente definiert:

Selection Timing legt fest, wann und ob überhaupt eine Kind-Aktion eines Cluster ausgewählt werden soll.

Selection Count Status überprüft, ob die ausgewählten Daten sinnvoll für die Aktion sind.

Selection Count legt die Anzahl der auszuwählenden Kinder fest.

3.3.2.8 Randomization Controls

Mit Hilfe dieser Kontrollelemente ist es möglich festzulegen, wann und welche verfügbaren Kindelemente der Cluster-Aktionen zur Laufzeit neu geordnet werden sollen (Reihenfolge). Es gibt zwei Elemente:

Randomization Timing bestimmt, wann ein Kind eines Clusters neu geordnet werden soll.

Randomize Children legt fest, ob es zu einer Änderung der Reihenfolge kommt.

Im Manifest lautet das entsprechende Element `<randomizationControls>`.

3.3.2.9 Delivery Controls

Mit Hilfe der Delivery Controls besteht für einen Entwickler von Lernobjekten die Möglichkeit, die Erfassung von Tracking-Informationen im so genannten Tracking Model zu beeinflussen. Er kann damit festlegen, ob ein Zugriff auf die Interaktion mit einer Activity vom LMS registriert werden soll oder weitere Werte bestimmen, die vom LMS nicht selbständig gesetzt werden dürfen.

Im Manifest werden Delivery Controls mit dem Element `<deliveryControls>` definiert; es umfasst drei Attribute: `tracked`, `completionSetByContent`, `objectiveSetByContent`.

3.3.3 Tracking Model

Mit der Einführung von Sequencing wurde zusätzlich zum RTE Data Model, ein weiteres Datenmodell spezifiziert, das Tracking-Modell oder auch Tracking Model. Dieses speichert dynamisch für jede Activity deren Zustand, der durch Interaktionen des Lernenden verändert wird. Diese so genannten Tracking-Informationen beinhalten beispielsweise die Anzahl der Zugriffe auf eine Activity, welche Activity gerade aktuell ist bzw. wie erfolgreich diese bearbeitet wird; alle Informationen zusammengefasst bilden das Tracking Model. Jedem Element sind anfangs Default-Werte zugeordnet, die zur Laufzeit durch Interaktionen des Lernenden verändert werden.

In Abbildung 3.17 wird der Zusammenhang zwischen dem Run-Time Environment Data Model und dem Tracking Model dargestellt - genauer betrachtet zwischen dem Activity Tree, speziellen Tracking-Informationen einer Activity, dem zugehörigen Content Object und dessen Runtime Data; teilweise werden Elemente vom RTE direkt auf Elemente des Tracking Models abgebildet.

Das Tracking Model definiert folgende Tracking-Status-Informationen:

Objective Progress Information: Beschreibt den Fortschritt eines Lernenden hinsichtlich eines Lernobjekts.

Activity Progress Information: Beschreibt den gesamten Fortschritt eines Lernenden samt allen Versuchen bezüglich einer Activity.

Attempt Progress Information: Beschreibt den Fortschritt eines Lernenden bezüglich des aktuellen Zugriffs auf eine bestimmte Activity.

Activity State Information: Für den Overall Sequencing Process (siehe Unterabschnitt 3.3.4) werden noch zusätzliche Statusinformationen für jede Activtiy gespeichert.

Global State Information: Hier werden Status-Informationen, die den Activity Tree betreffen, gespeichert. Diese Informationen werden auch während des

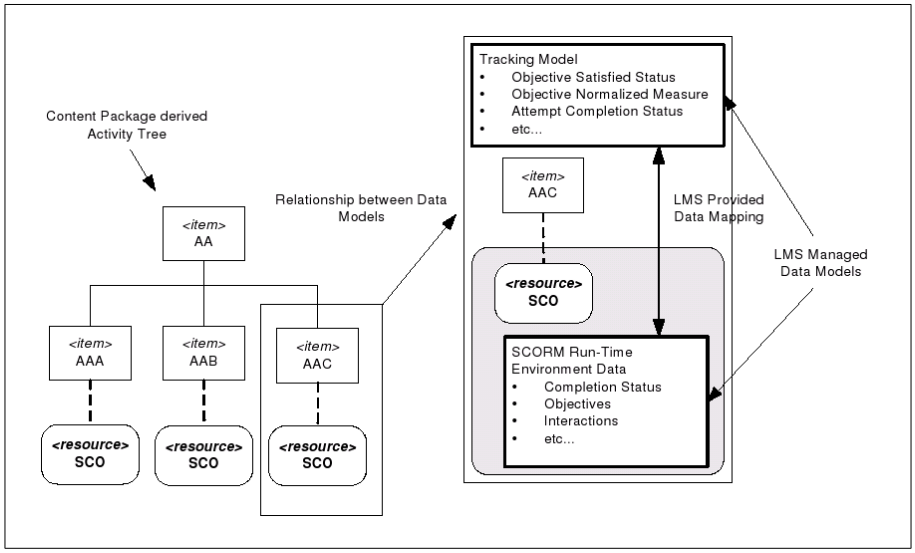


Abb. 3.17: Zusammenhang Tracking Model und RTE Data Model [ADL04d]

Overall Sequencing Process benötigt.

3.3.4 Overall Sequencing Process

Der *Overall Sequencing Process* beschreibt einen kompletten Steuerungsprozess für das Sequencing und definiert, wie die einzelnen Sequencing- Komponenten zusammenwirken bzw. sich verhalten:

Navigation Behaviour: Verarbeitet einen Navigation-Request und übersetzt diesen in einen Termination oder Sequencing Request.

Termination Behaviour: Beschreibt wie ein aktueller Versuch bei einer Aktion endet und der Aktionsbaum aktualisiert wird.

Rollup Behaviour: Beschreibt wie Tracking-Informationen für Cluster-Activities von Kind-Activities abgeleitet werden können.

Selection and Randomization Behaviour: Beschreibt wie Activities innerhalb eines Clusters während eines Sequencing-Requests behandelt werden sollen.

Sequencing Behaviour: Beschreibt wie ein Sequencing-Request im Activity Tree behandelt werden muss, um die nächste bereitzustellende Activity zu finden.

Delivery Behaviour: Überprüft, ob die Aktion bereitgestellt werden kann und sofern gültig wie diese in einem LMS behandelt werden soll.

Der Overall Sequencing Process beschreibt im Prinzip einen Workflow mittels Pseudocodes, die von Entwicklern für ein LMS exakt umgesetzt werden müssen. Er findet während einer *Sequencing Session* statt; also jenem Zeitraum zwischen dem Beginn eines Versuchs auf eine Wurzelaktivität bis zum Ende dieses Versuchs. Nach dem Start der Session befindet sich der Prozess in der so genannten *Sequencing Loop*. In Abb. 3.18 auf der folgenden Seite wird der Overall Sequencing Prozess dargestellt und folgend kurz erläutert:

Ablauf einer Sequencing Session/Loop

1. Der Lernende startet das LMS (z.B. mittels Login) und wählt einen Kurs (Content Organizazion) aus.
2. Das LMS startet einen Sequencing Prozess durch einen Start, Resume All- oder Choice Navigation Request.
3. Das Navigationsverhalten (Navigation Behaviour) übersetzt die Anfrage in einen entsprechenden Sequencing Prozess und verarbeitet diese.
4. An dieser Stelle beginnt die Sequencing Loop. Basierend auf einem Sequencing-Request, dem Tracking-Modell und dem Sequencing Definition Model wird eine Aktivität festgelegt, welche dem Lernenden präsentiert werden soll. Falls keine passende Activity gefunden wird, wird der Sequencing Prozess gestoppt bis ein anderer Navigation Request gestartet wird - es wird zu Schritt 9 gesprungen.
5. Der Lerner interagiert mit dem Content Object (Lerninhalt); währenddessen ruhen die Sequencing-Prozesse.
6. Das Inhaltsobjekt kann Werte rückmelden, die verschiedene Elemente des Tracking-Modells aktualisieren können.
7. Der Lernende, das Content Object oder das System erzeugen einen Navigation Request, wie zum Beispiel: Continue, Previous, Choose activity X, Abandon, Exit.
8. Das LMS informiert das Sequencing-Modul über das Navigationsereignis mittels eines Navigation-Request.

9. Das Navigationsverhalten (Navigation Behaviour) übersetzt den Request in einen Terminierungs-Request und einen Sequencing-Request. Falls diese Anfrage bedeutet, dass der Lernende den Versuch auf das Wurzelelement des Activity Trees beenden will, endet die Sequencing Session.

10. Falls das Content Object den Navigation Request durch seine Terminierung ausgelöst hat, besteht noch die Möglichkeit, Werte zu liefern, die das Tracking-Modell aktualisieren. An dieser Stelle wird das Rollup-Verhalten aufgerufen, um die Auswirkungen der Zustandsänderungen durch die Interaktionen des Lerners mit dem Inhaltsobjekt zu bestimmen. Das Rollup-Verhalten aktualisiert das Tracking-Modell für die Activity und für alle Vorgänger-Aktionen im Activity Tree.

11. Die Sequencing Loop wiederholt sich, beginnend mit Schritt 4 bis die Sequencing Session endet.

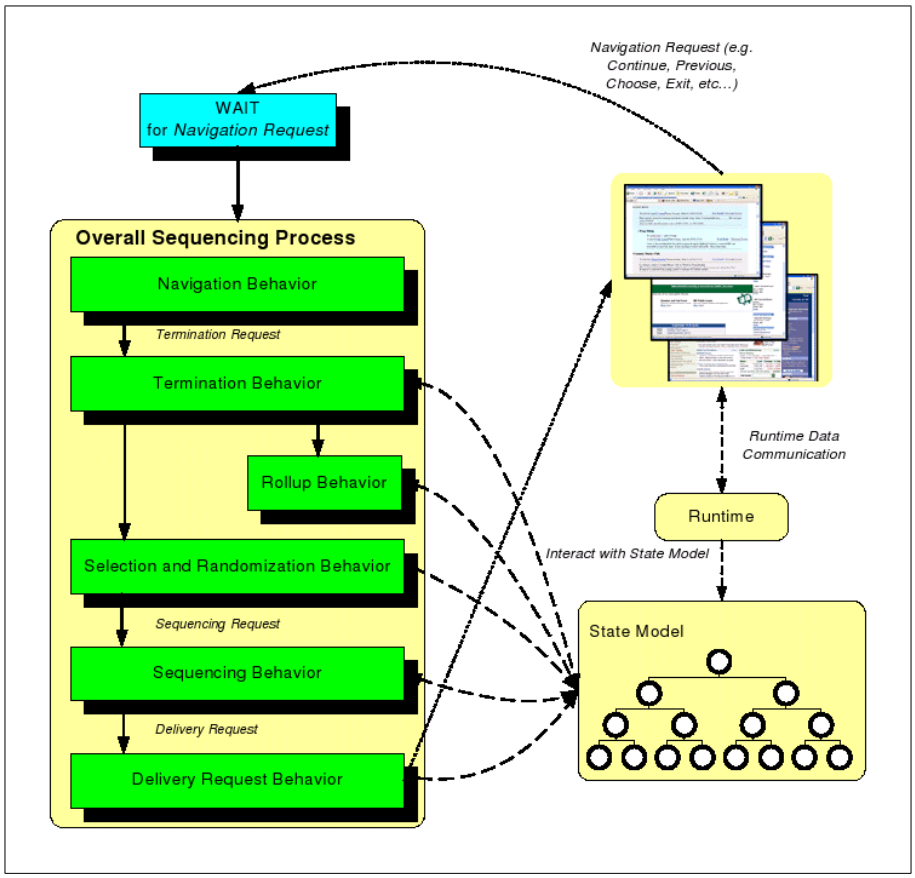


Abb. 3.18: Overall Sequencing Process [ADL04d]

3.3.5 Navigation Model

Navigationsanfragen (Requests) werden normalerweise über Benutzerschnittstellen – Navigationselemente wie Menüs oder Buttons –, die vom LMS bereitgestellt werden, ausgelöst. Um nun solche Schnittstellen direkt in Lerninhalte zu integrieren und Navigation Request direkt aus einem SCO zu erzeugen, wurde von ADL das Navigation Model hinzugefügt.

Das Modell ist nicht Bestandteil der IMS-Spezifikation sondern wurde von ADL zusätzlich erarbeitet. Es bezieht sich nur auf die Navigation zwischen verschiedenen Activities, die Navigation innerhalb eines Content Objects wird vom LMS nicht registriert bzw. gesteuert.

Das Modell besteht aus zwei Teilen:

Run-Time Navigation Data Model: Es können Navigation Requests von einem SCO zum LMS übertragen werden, weiters erfolgt eine Gültigkeitsüberprüfung des jeweiligen Navigation Request. Das Modell setzt sich aus zwei Elementen zusammen: `adl.nav.request` und `adl.nav.request_valid`. Mit Hilfe des ersten Elements kann das SCO den gewünschten Navigation Request zum LMS übertragen. Mit dem zweiten Element kann das SCO schon vorab die Gültigkeit eines Navigation-Prozesses überprüfen.

Presentation Navigation Model: indexSCORM!Navigation Model!Presentation Navigation Model Das Modell ermöglicht es festzulegen, welche Benutzerschnittstellen von einem LMS während der Anzeige des Content Objects nicht bereitgestellt werden sollen – redundante Schnittstellen werden damit vermieden.

In Tab. 3.2 auf der nächsten Seite werden die in SCORM spezifizierten Navigations-Ereignisse definiert mit den daraus resultierenden Navigation Requests.

3.3.6 Sequencing and Navigation SCORM Version 1.2

Sequencing and Navigation in der Version 1.2 war noch nicht als eigenständiges Buch sondern in der CAM Spezifikation unter dem Kapitel Sequencing and Na-

Ereignis	Auslöser	Verhalten	Request
Start	LMS	Sucht die erste verfügbare Activity im Activity Tree; das Ereignis wird typischerweise automatisch vom LMS ausgelöst, wenn ein Lernender einen neuen Versuch auf die Wurzelaktivität beginnt.	Start
Resume All	LMS	Setzt einen zuvor zurückgestellten Versuch auf die Wurzelaktivität fort; wird normalerweise automatisch vom LMS ausgelöst, wenn ein Lerner einen Versuch fortsetzt.	Resume All
Continue	LMS/SCO	Sucht die nächste Activity im Activity Tree.	Continue
Previous	LMS/SCO	Sucht die vorhergehende Activity im Activity Tree.	Previous
Choose	LMS/SCO	Wählt direkt eine bestimmte Activity aus.	Choice
Abandon	LMS/SCO	Beendet die aktuelle Activity, nicht jedoch eine etwaige Parent-Activity.	Abandon
Abandon all	LMS/SCO	Beendet die Wurzel-Aktivität des Baumes und alle aktiven Lernaktivitäten.	Abandon All
Suspend all	LMS	Stellen die Wurzel-Aktivität und alle Lern-Aktivitäten zurück; es wird jedoch keiner der Versuche beendet.	Suspend All
Unqualified Exit	LMS/SCO	Der aktuelle Versuch wurde ordnungsgemäß beendet; das Ende wurde nicht durch einen anderes Ereignis (Continue, Previous oder Choose) ausgelöst.	Exit
Exit All	LMS/SCO	Beendet den aktuellen Versuch auf die Wurzel-Aktivität und alle aktiven Lernaktionen.	Exit All

Tab. 3.2: SCORM 2004 Navigations-Ereignisse [ADL04d]

vigation zusammengefasst. Es beinhaltete eine noch recht einfache Form der Ablaufsteuerung von Kursen und wurde nur prototypisch realisiert; dies stellte sich auch als größter Nachteil dar. Hat ein Content-Entwickler die Ablaufsteuerung für einen Kurs festgelegt, kann diese nicht mehr geändert werden. Als Lernender besteht zwar eventuell die Möglichkeit, bestimmte SCOs via Navigationselementen – zum Beispiel Menüs – auszuwählen, die Struktur des Inhalts bleibt aber für jeden Anwender gleich.

In der Manifest-Datei wurde Sequencing mit dem Element `<adlcp:prerequisites>` realisiert. An jedes SCO konnten „Prerequisites“ gebunden werden – Vorbedingungen die erfüllt werden mussten, bevor ein Zugriff auf das SCO erfolgen konnte. Mit dem RTE-Daten-Modell wurde vom LMS überprüft, ob die Vorbedingungen erfüllt wurden.

Die Navigation wurde in SCORM 1.2 mittels Datentransfer-Befehlen durchgeführt. Es war nicht möglich, eine Navigation zwischen verschiedenen SCOs aus einem SCO heraus zu steuern.

3.3.7 Einschränkungen bzw. Unzulänglichkeiten

Die aktuelle Version der SCORM SN (Sequencing and Navigation) Spezifikation ist zwar bereits sehr ausgereift, dennoch gibt es Einschränkungen:

Die Spezifikation beschreibt wie die Struktur eines Kursinhalts als Activity Tree interpretiert werden kann, aber nicht, dessen Generierung, Darstellung oder Verwaltung. Ein Aktionsbaum muss nicht unbedingt statisch sein, seine Struktur und damit verbundenen Steuerungsinformationen können hingegen dynamisch verändert werden; wie Informationen für einen bestimmten Benutzer und Activitiy Tree persistent gemacht werden, ist in der Spezifikation nicht festgelegt.

Weitere Einschränkungen gibt es bei Anwenderschnittstellen: Es werden weder Aussehen oder Art der Schnittstelle noch Mechanismen für die Interaktion spezifiziert. Attribute wie Aussehen, Verhalten, Präsentation, Kontrollmechanismen von Anwenderschnittstellen sind nicht in der Spezifikation enthalten, es existieren lediglich Empfehlungen.

Das Steuerungsverhalten wird auch nur sehr unzureichend beschrieben: Ein

Auslösen von Navigationsanfragen oder -ereignissen wird nicht erläutert - lediglich Anfragen zum Starten einer Sitzung. Welche Navigationskontrollen jeweils sichtbar sind, wie sie interpretiert bzw. durch welche Navigationsereignisse sie ausgelöst werden, ist ebenfalls nicht näher spezifiziert. Weiters wird kein Steuerungsverhalten für Fälle definiert, in denen keine Activity dargestellt oder eine Fehlermeldung ausgelöst wird.

3.4 Fazit und Ausblick

Die von ADL entwickelte Spezifikation erfüllt in der aktuellen Version teilweise schon die von einem E-Learning-Standard geforderten Attribute wie hohe Wiederverwendbarkeit und Interoperabilität. Aufgrund von zahlreichen Kooperationen mit Industrie, privaten und öffentlichen Einrichtungen ist eine rasche Weiterentwicklung garantiert; gleichzeitig entstehen ständig SCORM-konforme Werkzeuge und Produkte.

Als positiv ist die äußerst ausführliche Dokumentation anzumerken. Weiters bietet ADL eine Referenzimplementierung der Laufzeitumgebung, Beispiele für Content-Packages und eine Conformance-Test-Suite an.

Trotz der positiven Aspekte wird noch vieles nicht von SCORM unterstützt, auch ist der Funktionsumfang noch gering. Nachdem die SCORM-Spezifikation nur ein technisches Dokument darstellt, fehlt es vor allem an didaktischen Standards.

Ein exakter Zeitplan für zukünftige Weiterentwicklungen von SCORM (Version 2.x) ist nicht genau festgelegt; voraussichtlich werden folgende Aspekte berücksichtigt:

- verbesserte Datenmodelle (Run-Time Data Model, Content Data Model)
- API-Unterstützung von SOAP (Simple Object Access Protocol)
- Agenten
- Learner Information Profiles
- verbesserte Assessments
- Bibliotheken zur Verwaltung von Inhalten

- Implementierung von SCORM-basiertem Intelligentem Tutoring und intelligentem Content
- Entwurf eines neuen Inhaltmodells (Content Model)
- Simulationen
- Anpassung an Darstellungsform / Multi Channel Content Delivery
- E-Learning-Pädagogik (Welche Verfahren sind für das Web am besten geeignet?)

Literatur: [IEE02], [ADL04a], [ADL04b], [ADL04c], [ADL04d], [Dec04], [Sch04b], [Glü04], [Tor04], [HG03], [Slo02], [Exn05], [Uni05], [Sch04c], [NP03], [ano05], [Gel05], [Wik]

Kapitel 4

Lern-Management-Systeme mit SCORM-Unterstützung

Bedingt durch die hohen Kosten bei der Erstellung von interaktiven Bildungsmedien hoher Qualität ist eine Wiederverwendbarkeit von großer Bedeutung. Insofern müssen schon bei der Erstellung der Medien bestimmte Standards eingehalten werden. Basierend auf dieser Tatsache haben schon früh viele Learning-Management-Systems versucht, aktuelle Standards zu integrieren bzw. das System darauf auszurichten oder zu erweitern.

Im folgenden werden kommerzielle sowie freie Plattformen kurz vorgestellt, weiters erfolgt eine Abschätzung der Vor- und Nachteile der einzelnen LMS und eine Überprüfung hinsichtlich verwendeter Standards unter besonderer Berücksichtigung der SCORM-Spezifikation. Abschließend werden noch Stand-alone-Applikationen zur Erstellung bzw. zum Editieren von SCORM-Inhalten überblicksmäßig betrachtet.

4.1 Einleitung

Der Markt von E-Learning-Produkten ist sehr unübersichtlich, die Anzahl an Lernplattformen wird je nach Definition in verschiedenen Studien auf bis zu 250 Stück angegeben, etwa 180 bis 200 sind proprietäre und 50-70 Open Source-Systeme. Firmen und öffentliche Bildungseinrichtungen stehen daher oftmals vor dem Problem, ein passendes System zu finden. Einerseits fehlt der Überblick über den Markt und

das Angebot an Systemen, andererseits ist es schwierig, sich für ein bestimmtes System zu entscheiden, ohne notwendige Erfahrungen aus der Praxis vorliegen zu haben.

In jüngster Zeit wurde im Auftrag des BMBWK (Bundesministerium für Bildung, Wissenschaft und Kultur)[1] eine Studie angefertigt, welche eine in mehrere Phasen gegliederte Evaluierung von Learning-Management-Systemen – vor allem für den Einsatz in Bildungseinrichtungen – vorsah.

Die Hauptkriterien der Evaluierung waren eine österreichische Rahmenlizenz und eine Web-basierende Lösung; Anwender sollten mit einem beliebigen Web-Browser am Lernprozess teilnehmen können und das gesamte Lernmanagement sollte ohne weitere Plugins via Browser bedienbar sein. Weiters sollten eine Benutzerverwaltung sowie eine Organisation (Erstellen, Erweitern, Löschen) von Content (Lerninhalte) und Kursen möglich sein. Das System musste ein entsprechendes Zugriffs- und Rechtemanagement hinsichtlich Rollen und Gruppen bieten. Auch sollte es LDAP-fähig bzw. entsprechend anpassbar sein. Eine weitere Forderung war die Unterstützung von zumindest zwei Sprachen (deutsch, englisch) und der Kommunikation innerhalb der Rollen mit entsprechenden Tools.

Betrachtet wurde bei der Evaluierung auch die synchrone und asynchrone Kommunikation und damit zusammenhängend das Tutoring. Als Empfehlung und somit die meisten Anforderungen erfüllend wurden u.a. die Produkte ILIAS (siehe Abschnitt 4.2.3.1 auf Seite 97) und WebCT (siehe Abschnitt 4.2.2.3 auf Seite 92) genannt.

4.1.1 Anforderungen

Für eine Analyse von Lern-Management-Systemen ist es zunächst notwendig, die zuvor genannten Evaluierungskriterien etwas genauer zu spezifizieren und grundlegende Eigenschaften bzw. Anforderungen an solche Systeme zumindest überblicksmäßig zu diskutieren:

Gruppenarbeit und Kommunikation: Das System sollte sowohl asynchrone als auch synchrone Kommunikation unterstützen; Beispiele hierfür sind Chat, E-

[1] http://www.bmbwk.gv.at

Mail, Mailing-Listen und Diskussionsforen. Weiters sollte es eine Bildung von Gruppen (Lehrende, Lernende, Tutoren, Administratoren, ...) ermöglichen und verbunden damit zumindest eine gruppenbasiertes, differenziert vergebbares Rechtemanagement unterstützen.

Didaktik/Kursverwaltung: Das LMS sollte in der Lage sein, verschiedene Veranstaltungstypen anzubieten. Für Veranstaltungen oder Kurse sollten spezielle Attribute berücksichtigt und dargestellt werden können. Es ergeben sich folgende Forderungen:

- Verwaltung von Veranstaltungen
- mittels Definition von Anforderungen und Lernzielen genauere Einschätzung über jeweilige Veranstaltung (Belegbarkeit, Ergebnis)
- Verwalten von Übungsaufgaben
- Durchführung automatisierter Selbsttests bzw. interaktiver Übungen und Tests (einzeln oder im Team), sowie eine automatische Beantwortung von Übungen
- Unterstützung einer zeitlichen Kopplung von Lehrmateralien und Freigabe benötigter Dokumente entsprechend dem Datum
- Verfügbarkeit sämtlicher (Veranstaltungs-)Daten mittels Archiv
- Online-Autofunktionen (z.B.: Vorlagen, Wizards, Rückmeldungen)
- Zulassen verschiedener Lehr- und Lernmodelle (z.B.: Lehrer-zentriert, Lerner-zenriert)
- Learning-Flow-Management (z.B.: Lernprozesssteurerung durch Lernpfade, Bookmarking)
- Modularisierung von Lehr- und Lerninhalten

Peronalisierung: Für einen Lernenden ist es von Bedeutung, das LMS entsprechend anzupassen. Zumindest ein Anzeigen der besuchten Veranstaltungen und Auswählen der Sprache sollten möglich sein. Weitere Forderungen sind:

- Erstellen von Notizen zu einem Dokument verbunden mit der Möglichkeit, ein Feedback zu geben
- Veröffentlichung allgemeiner oder persönlicher Ankündigungen seitens der Lehrer/Tutoren
- zumindest einfache, integrierte Kalenderfunktion
- individuelle Sprachauswahl für Informationsdarstellung

- Anzeigen personalisierter Informationen wie Lesezeichen, Ankündigungen oder ausgewählte Veranstaltungen für jeden angemeldeten Benutzer („persönlicher Arbeitsplatz“)

Technik: Um einen breiten Einsatzbereich und eine Kompatibilität mit anderen Lernplattformen zu gewährleisten, sind technische Anforderungen an ein LMS nicht zu vernachlässigen:

- standardisierte Import-/Exportschnittstelle für den Austausch von Daten
- Unterstützung verschiedener Serverbetriebssysteme, insbesondere Open Source Lösungen
- entsprechender Support (Response-Zeit, Erreichbarkeit, Sprache)
- Unterstützung von Standardobjekttypen (MS Office-Dokumente, PDF, Bildformate)
- Unterstützung von Internationalen E-Leraning Standards und Spezifikationen (z.B.: AICC, SCORM, IMS)
- Anpassbarkeit des Systems (Unterstützung von Templates, Corporate Identity)
- Erweiterbarkeit des Systems (Modularität, Offenheit für eigene Erweiterungen, Plugins, Makros)

Kosten: Bei Anschaffung eines Lern-Management-Systems ist auch eine vollständige Kostenabschätzung unumgänglich. Neben etwaigen Anschaffungskosten für das System selbst, können noch zusätzliche Kosten durch kontinuierliche Lizenzkosten und Einkauf nötiger Supportverträge entstehen. Die Höhe der Personalkosten hängt von der Administrierbarkeit und Delegierbarkeit der Aufgaben ab. Sachkosten entstehen beispielsweise durch Anschaffung neuer Hardware.

4.2 Evaluierung

Im folgenden werden kommerzielle (siehe 4.2.2) sowie Open Source (siehe 4.2.3) Anbieter von Learning-Management-System vorgestellt. Bezugnehmend auf mehrere Studien wurden jeweils die populärsten Lernplattformen ausgewählt. Im besonderen wird auch, sofern dokumentiert, das jeweilige System auf eine Unterstüt-

zung der SCORM-Spezifikation untersucht. Zuvor wird noch kurz das von ADL angebotene SCORM-Zertifizierungs-Verfahren vorgestellt.

4.2.1 SCORM Zertifizierung

Über das Academic ADL Co-Laboratory[2] in Madison, Wisconsin und das Naval Undersea Warfare Center (NUWC), Division Keyport[3], bietet ADL eine Zertifizierung von Lern-Management-Systemen und Lerninhalten nach SCORM an.

Dabei werden zwei Zertifzierungslevel angeboten: SCORM 1.2 und SCORM 2004. Überprüft werden das LMS selbst, SCOs, Metadaten für Content Aggregation, SCOs und Assets sowie das Content Package. Jeder Teil besitzt mehrere Level, die unterschiedlich strenge Vorschriften innehaben:

SCORM 1.2 konform

Learning-Management-System (LMS)

- LMS-RTE1
  - kann konforme Content Packages importieren und verarbeiten
  - kann konforme SCOs und Assets aufrufen
  - stellt die API mit allen Funktionen zur Verfügung
  - Alle notwendigen SCO-RTE1 + Mandatory Elements sind implementiert.
- LMS-RTE2
  - erfüllt LMS-RTE1
  - implementiert mindestens ein SCO-RTE1 Optional Element
- LMS-RTE3
  - erfüllt LMS-RTE1
  - Implementiert alle SCO-RTE1 Optional Elements

Shareable Content Object (SCO)

- SCO-RTE1

[2] http://www.academiccolab.org/certification/
[3] http://www-keyport.kpt.nuwc.navy.mil/ADL.htm

  - kann von einem konformen LMS ausgeführt werden
  - sucht und findet die API
  - verwendet zumindest die API-Funktionen `LMSInitialize()` und `LMSFinish()`
- SCO-RTE1+Mandatory
  - erfüllt SCO-RTE1
  - korrektes Lesen und Schreiben von mindestens einem Mandatory Element
- SCO-RTE1+Optional
  - erfüllt SCO-RTE1
  - korrektes Lesen und Schreiben von mindestens einem Optional Element
- SCO-RTE1+Mandatory+Optional
  - erfüllt SCO-RTE1+Mandatory und SCO-RTE1+Optional

Metadaten für Content Aggregation, SCO und Asset

- MD-XML1
  - well-formed XML Dokument
  - Alle geforderten Elemente sind enthalten.
- MD-XML1+Optional
  - erfüllt MD-XML1
  - Mindestens ein optionales Element ist enthalten.
- MD-XML1+Extensions
  - erfüllt MD-XML1
  - Mindestens eine Erweiterung ist enthalten.
- MD-XML1+Optional+Extensions
  - erfüllt MD-XML1+Optional und MD-XML1+Extensions

Content Package

- Wenn das Content Package in ein PIF-Paket gepackt ist, dann sollte diese Datei kompatibel zu PKZIP 2.04g sein.
- Das Manifest sollte „imsmanifest.xml“ heißen.
- Das Manifest und alle zugehörigen Kontrolldokumente sollten sich im

Wurzelverzeichnis der PIF-Datei befinden.

- Die Datei „msmanifest.xml“ sollte well-formed sein.
- Die Datei „imsmanifest.xml“ sollte entsprechend der IMS Content Packaging XSD v1.1.2 und der ADL Content Packaging XSD v1.2 gültig sein.
- Das Content Package sollte mindestens ein SCO oder Asset enthalten.
- Alle enthaltenen SCOs sollten mindestens SCO-RTE1 entsprechen.
- Alle in der Datei „imsmanifest.xml“ verwendeten Metadaten müssen korrekt sein.

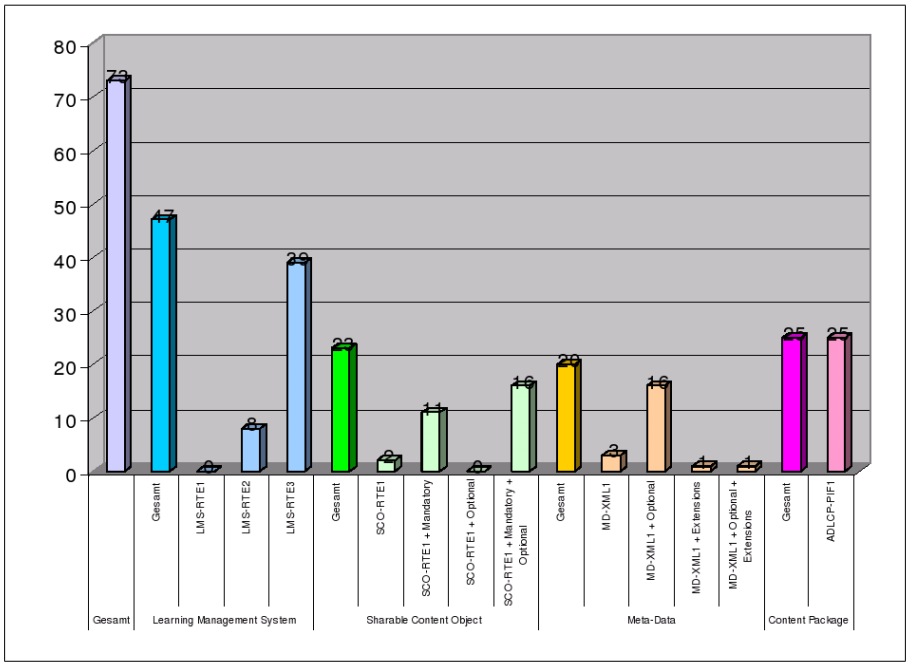


Abb. 4.1: ADL SCORM 1.2 zertifizierte Produkte (Stand März 2005) [Gel05]

SCORM 2004 konform

LMS

- LMS RTE 1.3.1
  - kann konforme SCOs und Assets aufrufen
  - stellt API mit allen Methoden zur Verfügung

  - implementiert alle Elemente des Datenmodells des Run-Time Environment und der Navigation
- LMS CAM 1.3.1
  - kann konforme Content Packages importieren und verarbeiten
  - kann Elemente des Datenmodells des Run-Time Environment nach den Vorgaben aus dem Manifest-File initialisieren
- LMS SN 1.3.1
  - implementiert das Sequencing Verhalten (Behaviour)
  - unterstützt alle Elemente des Datenmodells der Navigation
  - erfüllt die Anforderungen an das Navigation User Interface

SCO

- LMS RTE 1.3.1
  - sucht und findet die API
  - ruft erfolgreich zumindest die API-Methoden `Initialize()` und `Terminate()` auf
  - ruft erfolgreich die Data Transfer und Support Methoden auf, falls diese verwendet werden
  - Wenn Data Transfer Methoden verwendet werden, müssen die übertragenen Elemente den Anforderungen entsprechen.

Metadaten

- MD CAM 1.3.1
  - Die Metadaten müssen zu einem der Anwendungsprofile kompatibel sein:
    * Asset Metadata Application Profile
    * SCO Metadata Application Profile
    * Activity Metadata Application Profile
    * Content Organization Metadata Application Profile
    * Content Aggregation Metadata Application Profile

Content Package

- LMS CAM 1.3.1
  - Wenn das Content Package ein Content Aggregation Package Application Profile ist, muss das Manifest den entsprechenden Anforde-

rungen genügen.

- Dies gilt auch für Resource Package Application Profiles und die Erweiterungen für Sequencing and Navigation und Metadaten.

- LMS RTE 1.3.1
  - Das Content Package enthält mindestens ein SCO oder Asset.
  - Alle im Manifest enthaltenen SCOs erfüllen die Anforderungen an die Konformität von SCOs.

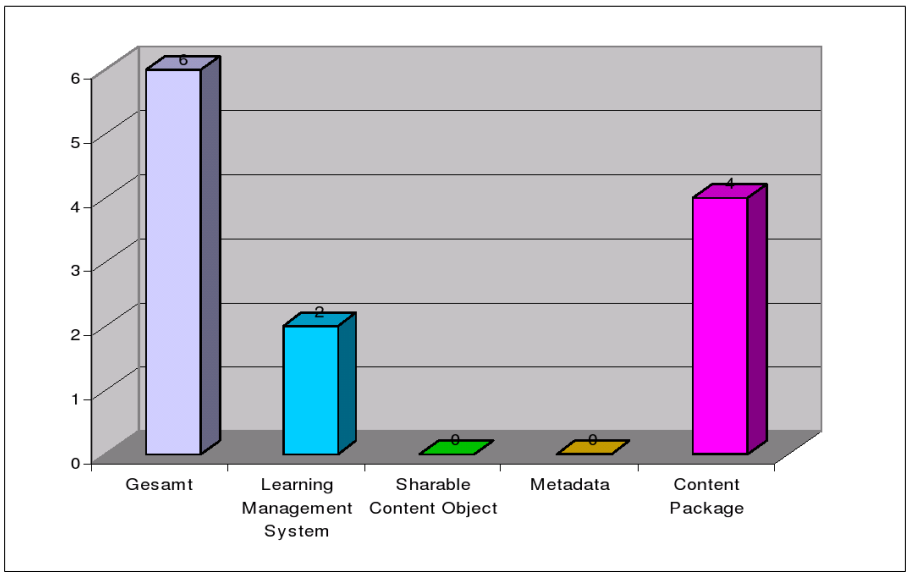


Abb. 4.2: ADL SCORM 2004 zertifizierte Produkte (Stand März 2005) [Gel05]

Eine gesamte Liste aller zertifizierten Produkte ist auf der ADL-Homepage[4] verfügbar.

Zur Überprüfung vor einer möglichen Zertifizierung bzw. auf SCORM-Konformität bietet ADL die so genannte SCORM 2004 Conformance Test Suite[5] an, um Selbsttests an SCORM Paketen, Metadaten und LMS durchführen zu können. Außerdem besteht die Möglichkeit des Downloads einer Referenzimplementierung einer SCORM-konformen Run-Time Umgebung[6].

[4] http://www.adlnet.org/scorm/certified/index.cfm
[5] http://www.adlnet.org/downloads/198.cfm
[6] http://www.adlnet.org/downloads/197.cfm

ADL SCORM Adopters

ADL bieten zusätzlich zur Zertifizierung ein Partnerprogramm für Einrichtungen und Produktanbieter , die SCORM unterstützen, an. Durch Nachweis der Konformität seiner Produkte über einen Selbsttest (SCORM 2004 Conformance Test Suite) und die Registrierung bei ADL kann jeder Anbieter in dieses Partnerprogramm aufgenommen werden. Eine Liste aller registrierter SCORM Adopters ist auf der ADL Homepage[7] ersichtlich.

4.2.2 Kommerzielle Lernplattformen

4.2.2.1 Blackboard

Allgemeine Informationen: Blackboard[8] ist neben WebCT einer der führenden Anbieter von Software für Online-Lernplattformen. Gegründet 1997 ursprünglich noch unter dem Namen „CourseInfo“ hat das LMS seinen Hauptsitz in Washington, DC. Blackboard besteht aus fünf Software-Anwendungen, die in zwei Suiten gebündelt sind: *Blackboard Academic Suite* und *Blackboard Commerce Suite*.

Funktionen: Diese beiden Suiten umfassen folgende Anwendungen und Funktionen:

- Das *Blackboard Learning System* ist eine Softwareanwendung für die Bereitstellung von Online-Lerninhalten. Diese beinhaltet Funktionen wie Kursverwaltung und Management, Content-Management, Kommunikation und Kollaboration, Analyse von Lernresultaten und System-Management.
- Das *Blackboard Content System* ist ein vollständig integriertes System mit zahlreichen Funktionen zum Speichern, gemeinsamen Verwenden und Veröffentlichen von Inhalten.
- Das *Blackboard Community System* verfügt über ein anpassungsfähiges Community-Portal, das akademische, kommerzielle, administrative und Community-Dienste über eine einzige integrierte Oberfläche online

[7] http://www.adlnet.org/scorm/adopters/index.cfm
[8] http://www.blackboard.com

zugänglich macht.

- Das *Blackboard Transaction System* unterstützt die Abwicklung von Geschäfts- und Zugangstransaktionen.
- *BbOne* ist ein System für bargeldlosen Zahlungsverkehr mittels Studentenausweis, das auch außerhalb des Campus benutzbar ist.

Architektur/Technik: Als Datenbank können sowohl der MS SQL-Server 2000 als auch Oracle 8 oder 9 eingesetzt werden. Unterstützte Server-Betriebssysteme sind Windows 2000 oder 2003 Server, Sun Solaris 8 oder 9 und RedHat Linux. Als Webserver können der Apache Webserver oder der Microsoft IIS (Internet Information Server) verwendet werden. Weiters wird noch Java 2 SDK benötigt. Das LMS ist auf allen gängigen Browsern lauffähig.

Standards: Es werden folgende Standards bzw. Spezifikationen unterstützt: IMS Metadata, IMS Content Packaging 1.1.2, IMS Question and Test Interoperability Specifications 1.2, IMS Enterprise Specification 1.01 und SCORM 1.2 (Level LMS-RTE3 zertifiziert).

Vorteile:

- übersichtliche Benutzeroberfläche
- leistungsfähiges „virtuelles Klassenzimmer“ mit synchronen Kommunikationsmethoden wie Chat und Whiteboard
- gute Unterstützung von Gruppenarbeit
- gute Teilnehmer- und Rollenverwaltung
- Einbindung von Inhalten auf CD-ROM möglich, um lange Ladezeiten bei großen Dateien zu vermeiden
- viele Sprachpakete
- einfach zu bedienen trotz umfangreicher Funktionalität
- Systemerweiterung mittels „Building Blocks“ (Hinzufügen neuer Funktionen bzw. Integration anderer Produkte)

Nachteile:

- sehr eingeschränkte Anpaßbarkeit des Layouts
- eingeschränkte Suchfunktionen
- vergleichsweise teuer

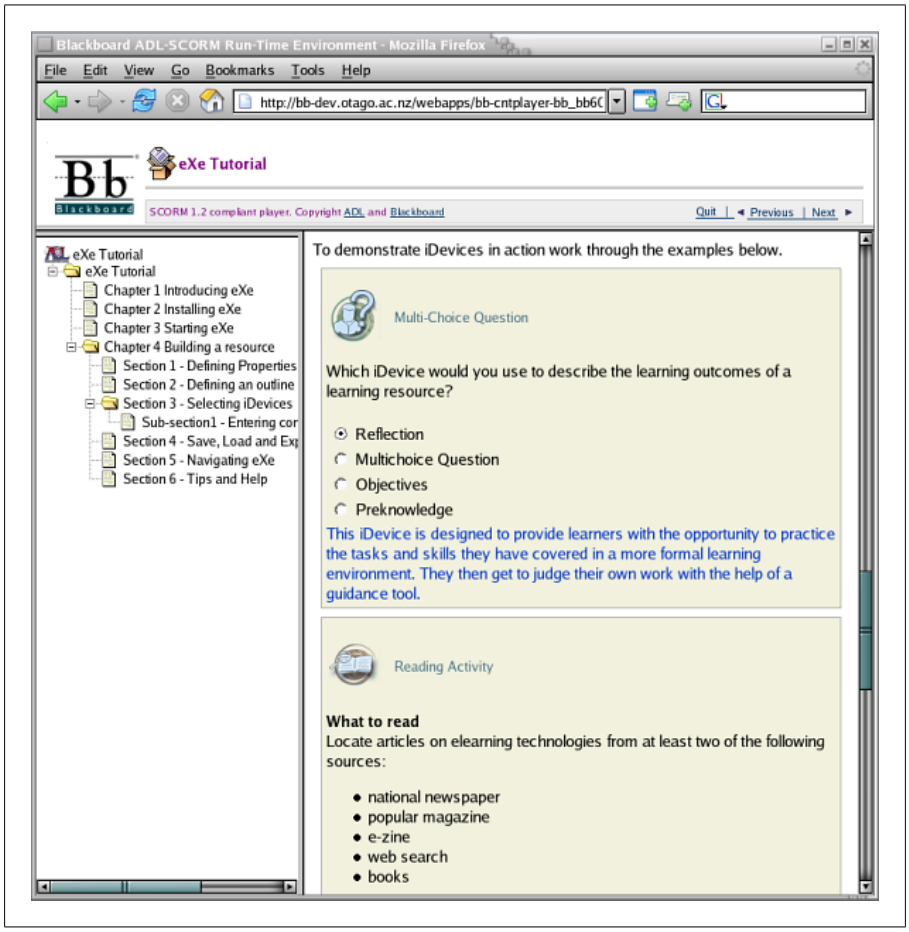

Blackboard ADL-SCORM Run-Time Environment - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://bb-dev.otago.ac.nz/webapps/bb-cntplayer-bb_bb6(
eXe Tutorial
SCORM 1.2 compliant player. Copyright ADL and Blackboard
Quit | Previous | Next
eXe Tutorial
eXe Tutorial
Chapter 1 Introducing eXe
Chapter 2 Installing eXe
Chapter 3 Starting eXe
Chapter 4 Building a resource
Section 1 - Defining Properties
Section 2 - Defining an outline
Section 3 - Selecting iDevices
Sub-section1 - Entering cor
Section 4 - Save, Load and Ex
Section 5 - Navigating eXe
Section 6 - Tips and Help
To demonstrate iDevices in action work through the examples below.
Multi-Choice Question
Which iDevice would you use to describe the learning outcomes of a learning resource?
Reflection
Multichoice Question
Objectives
Preknowledge
This iDevice is designed to provide learners with the opportunity to practice the tasks and skills they have covered in a more formal learning environment. They then get to judge their own work with the help of a guidance tool.
Reading Activity
What to read
Locate articles on elearning technologies from at least two of the following sources:
national newspaper
popular magazine
e-zine
web search
books

Abb. 4.3: Blackboard, SCORM-Kurs [exe05]

4.2.2.2 CLIX

Allgemeine Informationen: CLIX (Campus)[9] steht für *Corporate Learning and Information Exchange* und ist eine von der Firma IMC AG in Deutschland entwickelte E-Learning-Plattform. Es werden mehrere Produktvarianten angeboten: *CLIX Enterprise* für Unternehmen, *CLIX Campus* für Hochschulen und *CLIX Marketplace* für Bildungseinrichtungen und Online-Akademien.

Funktionen:

- Integration diverser Inhaltstypen (z.B. HTML, PDF, Bilder, Sound- und Video-Dateien)
- asynchrone und synchrone Kommunikationsformen wie E-Mail (Integration von Mailprogrammen via Schnittstelle), Diskussionsforen, Chat, News, FAQ-Listen, Blackboards und Dokumentenarchive
- Verwaltung von Kursen und Teilnehmern
- Tutor-Funktionen wie Monitoring
- Kontrolle des Lernfortschritts mittels Durchführung und Auswertung von Tests
- Gäste- und Adressbuch für jeden Nutzer
- Mediathek (für alle Nutzer zugängliche Mediensammlung)
- Übersicht über alle Nutzer in Form von Profilen
- Suchfunktionen
- hierarchisches Rollensystem (Student, Tutor, Kursadministrator, Portalgruppe, Administrator, Superuser)

Architektur/Technik: CLIX unterstützt Microsoft Windows NT, Windows 2000, Windows XP, Unix, Linux, AIX, Solaris, MacOS 9 und X. Als Datenbanksysteme werden Oracle 8i, IBM DB2 7.1 oder MS SQL Server verwendet. Für den Betrieb wird auch ein Application-Server benötigt, wie zum Beispiel: Macromedia JRUN, IBM Websphere, BEA Weblogic oder Apache Tomcat. Als Clients werden der Internet Explorer ab Version 5.0 und Netscape ab Version 6.1 empfohlen; bei Verwendung anderer Browser kommt es zu Funktionseinschränkungen. Weiters muss zumindest Java in der Version 1.2 verfügbar sein.

[9] http://www.im-c.de/homepage/

Das LMS basiert auf einer skalierbaren, mehrstufigen Web-basierten Client-Server Architektur. Sämtliche Module sind als serverseitige Java-Komponenten ausgeführt, wodurch CLIX (Corporate Learning and Information Exchange) plattformunabhängig und leicht auf vorhandene Hard- und Softwarekomponenten anpassbar ist.

Standards: Es werden folgende Standards bzw. Spezifikationen unterstützt: Dublin Core, IEEE LOM, AICC, SCORM. CLIX wurde mit der Umsetzung seiner SCORM 1.2 Schnittstelle mit der höchsten Stufe LMS-RTE3 zertifiziert; Import und Verarbeitung von SCORM-konformen Kursen funktonieren daher problemlos.

Vorteile:

- großer Funktionsumfang
- im Aufbau einer Lernumgebung sehr anpassungsfähig
- übersichtliche Benutzeroberfläche verbunden mit gradliniger Nutzerführung
- umfangreiche Personalisierungsfunktionen
- sehr gutes Rollen- und Rechtemanagement
- integrierte Kursvorlagen (Templates)
- große Auswahl an Fragentypen für die Erstellung von Tests
- Unterstützung der Versionsverwaltung von Inhalten
- Content-Converter zur Erzeugung von HTML-Dateien aus dem RTF-Format
- Unterstützung gängiger Standards

Nachteile:

- Installation und Administration sehr komplex, erfordert unter Umständen speziell geschultes Personal
- keine Autorenwerkzeuge integriert
- keine eigene Webseite und keine privaten Bookmarks für Nutzer möglich
- relativ hohe Anschaffungskosten

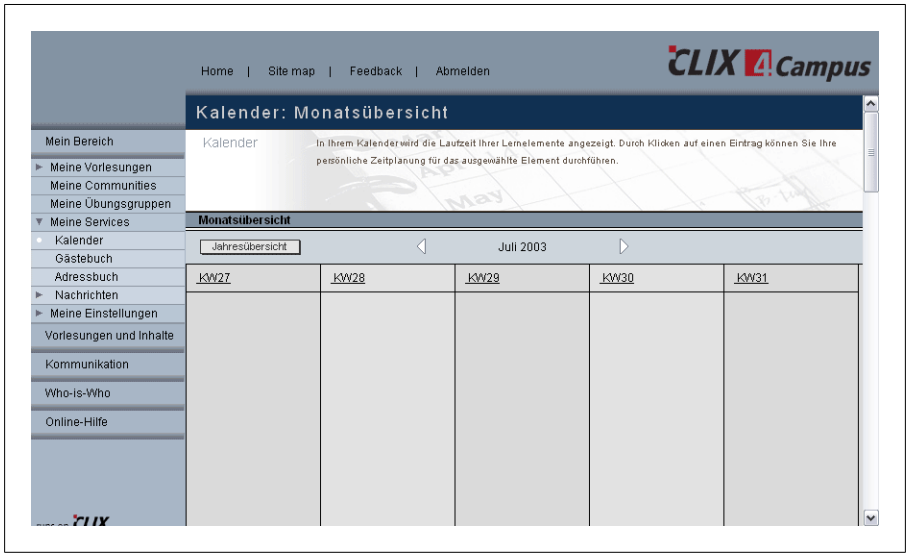


Abb. 4.4: CLIX, typische Startseite [est]

4.2.2.3 WebCT

Allgemeine Informationen: WebCT (Web Course Tools)[10] entstand 1995 aus einem Forschungsprojekt der University of British Columbia[11] in Vancouver, Kanada und gehört heute zu den weltweit führenden Systemen für die Erstellung und die Verwaltung von Online-Lernumgebungen im akademischen und im Weiterbildungsbereich. Es wird von mehr als 10 Millionen Studenten an mehr als 2500 Institutionen in 80 Ländern der Welt eingesetzt. Die letzte Version *WebCT Vista Academic Enterprise Edition* löst die bis dahin aktuelle *WebCT Campus Edition* ab. WebCT Vista ist eine komplette Neuentwicklung der Campus Edition (CE) basierend auf Java und setzt auf dem BEA Weblogic Applikationsserver[12] auf. Beide Version werden zur Zeit noch parallel angeboten.

Funktionen:

- persönlicher Desktop „my WebCT“ und Bookmarks
- Medienbibliothek: Integration vielfältiger Materialien und Dateiformate wie Texte, Bilder, Video- und Audio-Dateien, komplexe Gleichungen, Bilddatenbanken, Indizes, Glossare und bereits vorhandene Kursmate-

[10] http://www.webct.com
[11] http://www.ubc.ca
[12] http://www.bea.com

rialien

- Editieren und Verwalten von Online-Texten mit Hilfe eines integrierten HTML (Hypertext Markup Language)-Editors
- Lernmodule (Strukturierung und Organisation von Lehrinhalten)
- asynchrone Kommunikationsformen: E-Mail, Diskussionsforum (entspricht einer Newsgroup), Bekanntmachungen (Ankündigungen an Studierende)
- synchrone Kommunikation: Echtzeit-Chats (virtuelle Sprechstunden) und interaktives Whiteboard
- Unterstützung der Kommunikation und des Datenaustauschs sowie der Ergebnispräsentation innerhalb einer Gruppe
- Syllabus: Vermittelt Studenten allgemeine Informationen zu Lehrveranstaltungen
- Kalenderfunktion mit drei Ebenen (universitätsweit, kursbezogen und persönlich)
- „Grade Book“: Automationsunterstütze Notenverwaltung (Leistungsübersicht)
- Verwaltung von Kursen und Teilnehmern
- Evaluierungs-Tools (Aufgabenverwaltung, Auswertung von Tests, Feedback)
- Überprüfung studentischer Leistungen (Lernfortschritt) durch Tests und Aufgaben, Fortschritts-Ermittlung durch Selbst-Tests, Online- Studienbuch
- Definition von (Benutzer-) Rollen wie Administrator, Helpdesk-User, Designer, Tutor oder Student

Architektur/Technik: WebCT Campus unterstützt alle gängigen Browser, Java muss für einige Zusatzanwendungen installiert sein. Als Server können sowohl Windows 2000 (Advanced) Server als auch Red Hat Linux oder Sun Solaris (8,9) eingesetzt werden. Weiters werden Perl 5.6.1 und Apache ab Version 2.0 benötigt.

Während die Campus Edition eine Sammlung von serverseitigen, editierbaren Perl-Skripts darstellt, besteht WebCT Vista aus einer vierstufigen Architektur: Java 2 Platform Enterprise Edition (J2EE), Oracle 9i Datenbank, Oracle

Internet File System (IFS), BEA WebLogic Enterprise Server.

Standards: WebCT Vista ist SCORM 1.2 (Level LMS-RTE3) zertifiziert und verfügt über ein eigenes SCORM-Modul, womit es möglich ist, SCORM-kompatible Inhaltspakete zu importieren. Nach erfolgreichem Import wird der SCORM-Kurs auf der Benutzeroberfläche angezeigt, einschließlich eines Inhaltsverzeichnisses und der Navigationsschaltflächen. Mit dem integrierten Modul können folgende Schritte ausgeführt werden:

- Festlegen oder Bearbeiten von Freigabekriterien zur Steuerung der Verfügbarkeit von SCORM-Kursen
- Festlegung einer Benotung
- Ausführen von Datenberichten (Anzeige welche Studenten einen Kurs angezeigt haben, Kursfortschritt, Benotung, Vorschaumöglichkeit, Abfrage über Verfügbarkeit, Gesamtdauer der Anzeige eines Kurses durch einen Studenten)

Weiters werden IMS Content Packaging Specification 1.1.2, IMS Question & Test Interoperability Specifications 1.2, IMS Enterprise Specification 1.01, IMS Metadata v.1.2.1/IEEE LOM und Microsoft LRN 2.0. unterstützt.

Vorteile:

- vielfältige Erfahrungen mit dem Einsatz an Hochschulen und an deren Bedürfnisse angepasst
- relativ einfache Bedienbarkeit, intuitive Benutzeroberfläche, keine Programmierkenntnisse für den Tutor erforderlich
- flexibles, vergleichsweise günstiges Lizenzmodell
- leistungsstarke (Kommunikations-)Tools
- geringe technische Voraussetzungen, sehr stabiles System
- Einbindung von Inhalten auf CD-ROM möglich, um lange Ladezeiten bei großen Dateien zu vermeiden
- Unterstützung aller gängigen Standards und Spezifikationen
- gute Dokumentation
- große Funktionsvielfalt
- Unterstützung großer Benutzerzahlen (>30000)
- hohe Skalierbarkeit

Nachteile:

- Schwächen in der Kurs- und Ressourcenverwaltung
- geringe Auswahl an Fragetypen für die Erstellung von Tests
- diverse Browserfunktionen wie Bookmarks oder Vor- und Zurück-Buttons funktionieren unter Umständen gar nicht oder nur in eingeschränkter Form
- Bedienung diverser Assistenten ist teilweise sehr umständlich
- umständlicher Upload-Mechanismus (immer nur maximal eine Datei)
- unübersichtliche (Baum-)Struktur in Diskussionsforen

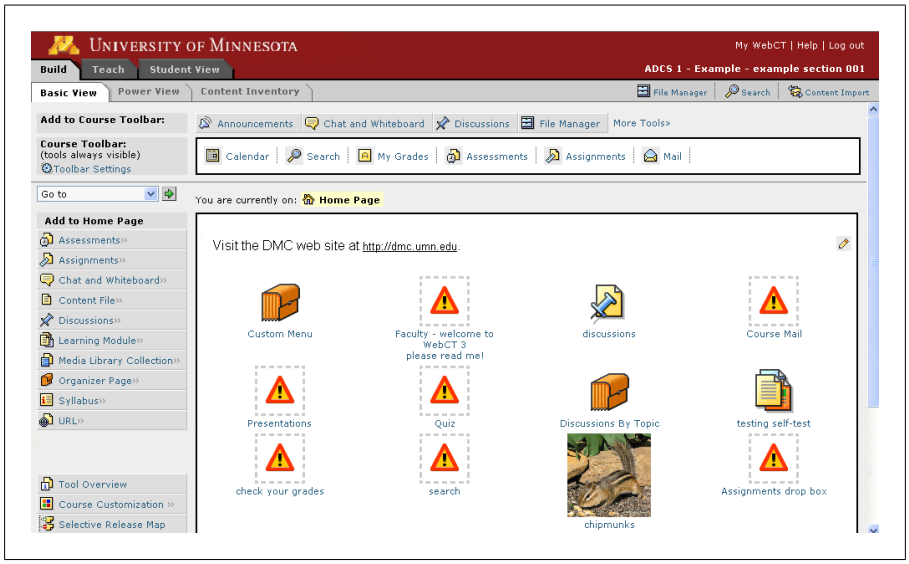


Abb. 4.5: WebCT Vista, Interface für das Erstellen eines Kurses [min]

4.2.3 Open Source Lernplattformen

Open Source Software wird als Alternative zu kommerziellen Produkten zunehmend beliebter, selbst im E-Learning-Bereich. Oftmals wird als einziger Vorteil die freie Verfügbarkeit gesehen, dadurch wird jedoch der Eindruck erweckt, dass damit E-Learning zum „Nulltarif“ möglich wird. Das Merkmal *frei* bezieht sich vielmehr auf eine Freiheit im Umgang mit der Software im allgemeinem und dem frei zugängliche Sourcecode. Es besteht dabei die Möglichkeit, diesen zu ändern und

verändert weiterzuverbreiten. Ein weiterer Grund für die Verbreitung ist die Tatsache, dass Open Source Software im Bildungsbereich – vor allem an Hochschulen – nicht nur eingesetzt sondern auch entwickelt, modifiziert und weiterentwickelt wird. Es fehlen oft einfach die erheblichen finanziellen Mittel für den Einsatz kommerzieller Software.

Open Source unterliegt in vielen Fällen den Bestimmungen der GNU General Public License (GPL), eine von der Free Software Foundation[13] herausgegebene Lizenz für die Lizenzierung freier Software. Diese soll garantieren, eine Software nutzen zu können, weiterzugeben und zu verändern. Weiters soll sichergestellt werden, dass die Software für alle Benutzer frei zugänglich ist. Folgende Auflagen gehören unter anderem zur Lizenz:

- Es gibt keinerlei Gewährleistung oder Garantie.
- Jede Kopie der Software muss einen entsprechenden Copyright-Vermerk sowie einen Haftungsausschuss enthalten.
- Veränderungen im Source müssen entsprechend gekennzeichnet werden.
- Änderungen müssen dokumentiert und mit einem Datum versehen sein.
- Ohne Annahme der Lizenz darf nichts geändert und weiterverbreitet werden.

In diesem Zusammenhang sollte man jedoch nicht die GPL und Open Source Software gleichsetzten; letztere bezieht sich auf die Open Source Initiative (OSI)[14], deren vorrangiges Ziel es ist, die Akzeptanz von Open Source in der Wirtschaft zu steigern, die bis dahin durch die restriktiven Bedingungen der GPL eingeschränkt wurde. Die OSI hat daher eine eigene Open Source Definition veröffentlicht:

1. uneingeschränkte Weiterverbreitung der Software
2. Das Programm enthält den Quellcode, der in einer verständlichen Form verbreitet werden muss.
3. Die Lizenz muss Modifikationen, Weiterentwicklungen und ihre Distribution zu denselben Lizenzbestimmungen erlauben.
4. Unversehrtheit des Quellcodes von Autoren

[13] http://www.fsf.org
[14] http://www.opensource.org

5. keine Diskriminierung von Personen oder Gruppen
6. keine Einschränkungen von Einsatzmöglichkeiten der Software
7. Bei der Verbreitung der Lizenz gelten die zum Progamm gehörenden Rechte.
8. Die Lizenz darf nicht für ein bestimmtes Produkt gelten.
9. Die Lizenz darf andere Software nicht einschränken.
10. Die Lizenz muss technologisch neutral sein.

4.2.3.1 ILIAS

Allgemeine Informationen: ILIAS[15] steht für *Integriertes Lern-, Informations- und Arbeitskooperations-System* und ist eine der populärsten und ausgereiftesten Open Source Online-Lernplattformen. Entwickelt im Rahmen des VIRTUS-Projekts[16] an der Universität Köln unterliegt es der GNU General Public License (GPL) und kann kostenlos genutzt werden.

Mittlerweile beteiligen sich mehrere Hochschulen und Institutionen an der Weiterentwicklung; besonderes Augenmerk wird dabei auf einen Einsatz offener Schnittstellen und Standards gelegt, um damit größtmögliche Interoperabilität und Nachhaltigkeit zu gewährleisten. Generelles Ziel von ILIAS ist es, Kosten für die Nutzung neuer Medien in Aus- und Weiterbildung möglichst niedrig zu halten und gleichzeitig höchstmöglichen Einfluss der Anwender auf die Gestaltung und Entwicklung des LMS zu gewährleisten.

Funktionen: Die aktuellste Release-Version 3.3.4 verfügt über folgende Module und Funktionalitäten:

- rollenbasiertes RBAC (Role Based Access Control) zur modularen Vergabe von Zugriffsrechten an verschiedene Anwender und Gruppen
- „Persönlicher Schreibtisch“ als zentraler Arbeitsplatz jedes Benutzers mit Rubriken für eigene Lernmodule, Kurse, Gruppen, Foren und Bookmarks sowie einer persönlichen Visitenkarte und einem Kalender
- Magazin für alle Lerninhalte, Foren und Arbeitsgruppen, die man benutzen und sich auf seinen Schreibtisch legen kann

[15] http://www.ilias.uni-koeln.de
[16] http://www.virtus.uni-koeln.de

- Integrierte Autorenumgebung: Erlaubt das Anlegen und Bearbeiten von Lernmodulen oder Digitalen Büchern und verfügt über einen WYSIWYG -orientierten Editor, wodurch Texte online erstellt und Bilder oder Multimedia-Objekte eingebunden werden können.
- Forensystem für beliebig viele Diskussionsforen
- Nachrichtensystem zum Versand von internen Mails und von E-Mails an externe Adressen samt Attachments
- Gruppensystem, das kooperatives Arbeiten und Lernen in Teams unterstützt. Es gibt vier Systemgruppen: Autoren, Lehrer, Gäste und Administratoren.
- zusätzliche Funktionen und Tools, wie einen Chat und Online-Tutor
- Noch im Entwicklungsstadium befindet sich ein Assessment-Tool für die Erstellung von Tests, ein Kursmanagementsystem und Übungen.

Architektur/Technik: Das bevorzugte Betriebssystem ist Linux (Installationsroutinen) bzw. Unix; ein Betrieb auf anderen Plattformen wie Windows und MacOS X ist möglich, der Support hierfür ist aber nur eingeschränkt vorhanden. Für den Betrieb werden der Webserver Apache (ab Version 1.3), die Datenbank MySQL und die Skriptsprache PHP benötigt. Die aktuellsten Versionen von ILIAS (3.3.4 bzw. 3.5.0_beta2) benötigen PHP4, idealerweise als Apache-Modul konfiguriert.

Das Grundgerüst der Seiten besteht aus PHP, HTML-Templates und CSS (Cascading Style Sheets). Einige zusätzliche Funktionsmodule wie die XML-Verarbeitung und Chat-Funktion benötigen Java.

Standards: Es werden mehrere Standards unterstützt, wobei Darstellung (Navigation), Validierung (abhängig von installierter Java-Version) und Funktion großteils Browser-abhängig sind; mit Mozilla und Firefox treten am wenigsten Probleme auf.

Es können AICC konforme Lernmodule importiert werden, der LOM Standard wird ebenfalls unterstützt. Die SCORM-API ist fast ausschließlich für die Version 1.2 implementiert, neuere Versionen wie SCORM 2004 können importiert jedoch nicht verarbeitet werden. Die Validierung bei Import eines Content Packages erfolgt serverseitig mittels Java, deaktiviert man diese, können auch IMS Content Packages importiert werden. Die API arbeitet Clientseitig mittels eines Java-Applets , das mit serverseitigen PHP-Scripts kommu-

niziert.

Vorteile:

- kostenlos
- Open Source: Jeder kann potentiell Mitentwickler werden, Fehler verbessern, Weiterentwicklungen betreiben und Lösungen weitergeben.
- ursprünglich an einer Hochschule entwickelt, daher auch an den Bedürfnissen von Hochschulen ausgerichtet
- vielfältige Erfahrungen im Einsatz an Hochschulen
- Unterstützung von Metadaten auf allen Inhaltsebenen
- Verwendung von E-Learning-Standards/Spezifikationen
- kontextsensitive Hilfe
- viele Systemsprachen integriert

Nachteile:

- insgesamt nicht ganz so leistungsfähig wie einige kommerzielle Produkte (Einschränkung z.B. bei Fragetypen und Fragenpooling)
- komplexer, lang dauernder Installationsprozess (bis zu 10 Stundent)

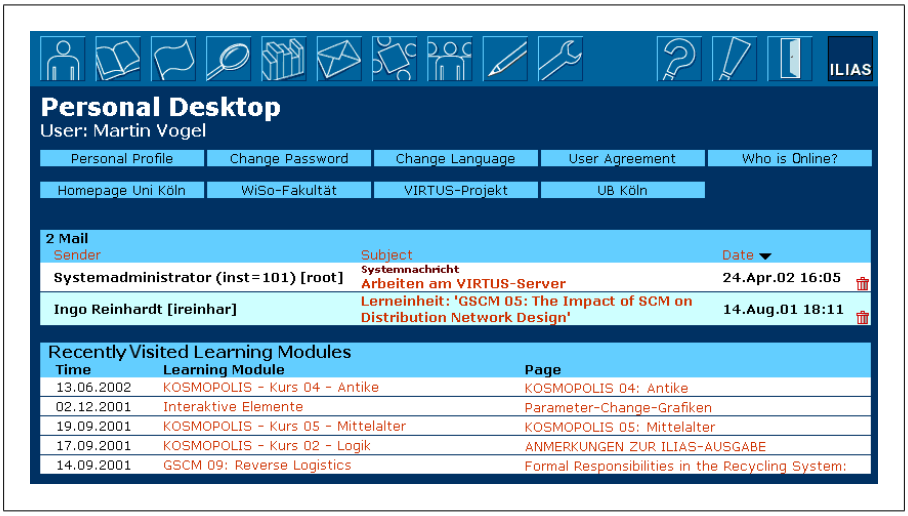


Abb. 4.6: ILIAS, Startseite („Persönlicher Schreibtisch“) [ili05]

4.2.3.2 ATutor

Allgemeine Informationen: ATutor[17], ein Web-basiertes LCMS, wird an der University of Toronto[18] in Kanada entwickelt. Der Entwicklungsschwerpunkt liegt bei einer einfachen Benutzung des Systems, mit besonderer Berücksichtigung von behinderten Menschen (z.b. Sehbehinderung). Hierfür wurden die WCAG (Web Content Accessibility Guidelines) 1.0 der W3C WAI (Web Accessibility Initiative)[19] implementiert, die als grundlegender Standard für eine barrierefreie Webseitengestaltung gelten. Weiters wurde in der aktuellsten Version das Datenbanksystem TILE (The Inclusive Learning Exchange)[20] integriert, wodurch Lerneinheiten für andere Kurse wiederverwendet werden können. Die Plattform unterliegt der GPL Lizenz und ist aktuell in der Version 1.5.1 verfügbar.

Funktionen:

- Unterstützung mehrer Sprachen (bisher 17 Sprachpakete, Standardsprache ist Englisch)
- Personalisierte Oberfläche: Anordnung und Inhalt des Menüs sowie der Navigation können verändert werden.
- Kommunikations-Tools: Lernende können mittels integriertem Mailsystem, Diskussionsforum oder Chat miteinander kommunizieren.
- Lokales Speichern von Inhalten für Offline-Benutzung
- Test-Manager: Erstellen von Fragen wie Multiple Choice (wahr/falsch), Vergleiche (wichtig/unwichtig) und offene Antworten (1 Wort bis 1 Seite) durch den Kursleiter. Tests werden automatisch, manuell oder gar nicht benotet.
- Dateimanger
- Kursverwaltung
- Import/Export bestehender Inhalte
- rollenbasierende Benutzerverwaltung: Student, Instruktor für Kurse, Administrator

[17] http://www.atutor.ca/
[18] http://www.utoronto.ca/
[19] http://www.w3.org/WAI/
[20] http://inclusivelearning.ca

- Tracking-System: Möglichkeit, den Weg eines Lernenden durch den Lehrstoff zu verfolgen (Dokumentation von Seitenverlauf und Verweildauer)
- „ATalker“: Wiedergabe von Lehrinhalten als gesprochenes Wort für sehbehinderte Teilnehmer
- Statistik-Tool
- TILE Repository-Suche
- WYSIWYG Content-Editor
- „Accessibility Checker“: Überprüfung von Lerninhalte auf Barrierefreiheit

Architektur/Technik: Die technische Realisierung erfolgt auf Basis einer AMP Plattform: Apache (empfohlen 1.3.x), MySQL (4.0.12 oder höher) und PHP (ab 4.2.0). Es werden alle gängigen Betriebssysteme und Browser unterstützt.

Standards: ATutor kann Lerninhalte in den Formaten IMS 1.2 CP (Content Packaging), SCORM 1.2 CP und SCORM 2004 importieren; AICC wird (noch) nicht unterstützt. Bei Import eines Kurses werden alle in der Manifest-Datei referenzierten HTML-Dateien in einer Datenbank gespeichert; dabei wird nur der HTML-Code innerhalb der `<body>`-Elemente gespeichert und für die Darstellung in ATutor eigenem HTML-Code eingefügt. Links zwischen HTML-Seiten des Lernmoduls oder auch Cascading Style Sheets gehen dabei verloren. Die SCORM-typische Framestruktur bereitet auch noch Probleme.

Inhaltspakete, importierte oder selbst erstellte und einzelne Seiten daraus, können als SCORM-kompatibles Paket exportiert werden. Die SCORM-API für den Austausch von Daten zwischen Lernobjekten und dem LMS wird nicht unterstützt.

Vorteile:

- kostenfrei verfügbar
- keine Installation spezieller Software nötig
- gesetzeskonforme, barrierefreie Bedienung
- Standard-konformer Export und Import
- Anpassung der Optik möglich
- viele Systemsprachen integriert

Nachteile:

- Einschränkungen bei komplexerem Einsatz und praxisnaher, effektiver Verwendung (Anpassung notwendig)
- Unübersichtliches Layout (zu große Schriftarten bei bestimmten Auflösungen)
- Menüstruktur nicht intuitiv benutzbar

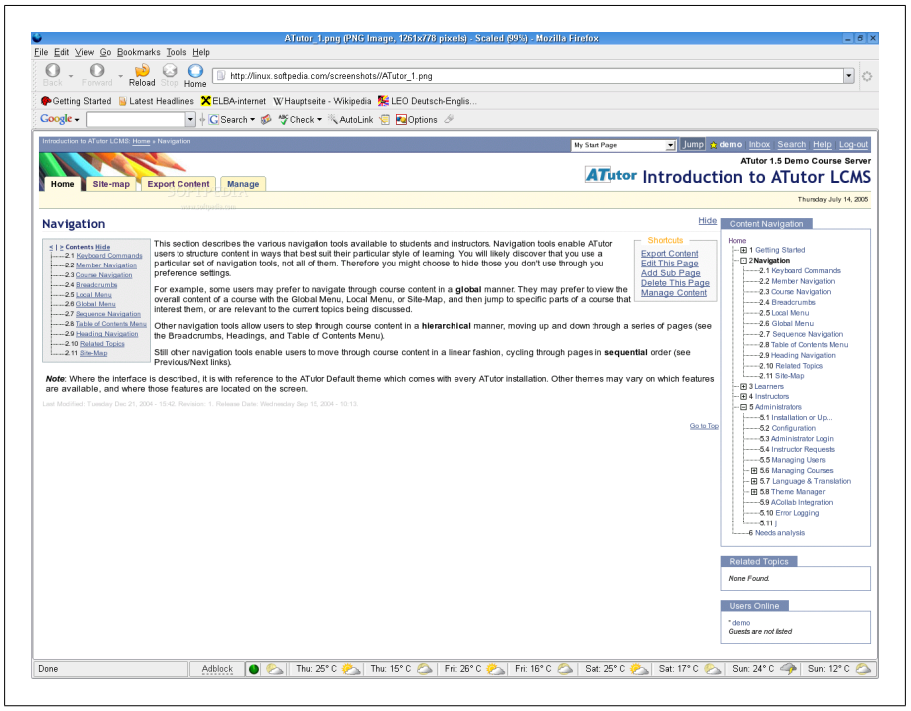


Abb. 4.7: ATutor, Startseite [sof]

4.2.3.3 Moodle

Allgemeine Informationen: 1999 begann der Australier Martin Dougiamas mit der Entwicklung des Learning-Management-Systems Moodle (Modular Object Oriented Dynamic Learning Environment) [21], um Bildungseinrichtungen bei der Erstellung von hochqualitativen Online-Kursen zu unterstützen. Vor-

[21] http://moodle.org

rangiges Ziel ist es, eine konsequente Umsetzung von Erkenntnissen der konstruktivistischen Pädagogik.

Im August 2000 wurde nach langwierigen Evaluierungsverfahren die erste endgültige Moodle-Version veröffentlicht. Mittlerweile ist eine stark engagierte Moodle-Community entstanden, die durch verteilte Entwicklung, parallel an mehreren Teilprojekten arbeitet. Moodle ist dadurch streng modular aufgebaut, neben grundlegenden Funktionen und Standard-Modulen, lässt sich das System, einfach um weitere Module erweitern bzw. anwendungsspezifisch anpassen. Das LMS umfasst folgende Funktionen:

Funktionen:

- Assignment-Tool: Festlegen von Tasks
- Unterstützung von Gruppenarbeit (z.B. in Foren oder Wikis)
- Wiki-Funktion
- Diskussionsforen
- synchrone Echtzeitkommunikation mittels Chat
- Überblick über Aktivitäten der Studenten
- Definition von Rollen (Administrator, Kursersteller, Trainer, Teilnehmer, Gast)
- Lerntagebuch
- Abstimm- und Umfragefunktion
- Sprechstunde mit Trainer
- Glossar
- Tests: Multiple Choice (einfach/mehrfach, wahr/falsch), Kurzantworten, numerische Antworten und Berechnungen, Zuordnungen, Beschreibungen und Lückentexte. Zur Auswertung stehen umfangreiche Statistiken zur Verfügung.
- Dialog: Privater Kommunikationsbereich zwischen Benutzer
- Umfragen: Vier Umfragetypen mit vordefinierten Fragen
- Lektionen: Erstellen von Lernpfaden, Import/Export von Fragen möglich
- Kalenderfunktion
- Anwesenheitskontrolle

Architektur/Technik: Moodle wurde unter Linux entwickelt und benötigt für den Betrieb eine AMP-Umgebung; PostgreSQL als Datenbank wird ebenfalls unterstützt, weitere Datenbanktypen erfordern eine manuelle Einrichtung der Tabellen. Das System ist auch unter Windows, MacOS X oder Netware lauffähig, sofern ein PHP-Interpreter ansprechbar ist. PHP sollte zumindest in der Version 4.1.0 vorliegen, JPG und PNG unterstützen.

Standards: SCORM 1.2 wird seit der Version 1.5.2+ vollkommen, SCORM 2004 voraussichtlich ab Version 1.7 unterstützt. Ein Import funktioniert bereits für beide Versionen. Seit kurzem wird auch der Import von AICC Paketen unterstützt, hierbei können jedoch noch Probleme auftreten.

Moodle erstellt für jeden importierten Kurs ein eigenes Verzeichnis im Datenordner, in welches die gepackten Pakete geladen werden. Die Darstellung von SCORM-Kursen erfolgt wie üblich mittels Frames, die importierten Seiten bleiben unverändert, darin enthaltene CSS und Links sind anwendbar.

Vorteile:

- kostenlos
- große und aktive Community
- viele Systemsprachen integriert (bis zu 34)
- Kurse können als Zip-Dateien gepackt und auf jedem anderen Moodle-Server importiert werden
- geringe Einarbeitungszeiten sowohl auf Seiten des Lehrenden als auch auf Seiten der Lernenden
- automatische Datensicherung
- Anpassung der Darstellung möglich
- einfache Installation
- einfaches, effizientes, kompatibles Browserinterface
- Aufbau mittels Module, dadurch einfache Erweiterungs- und Modifikationsmöglichkeiten
- großes Augenmerk auf Sicherheit (Datenvalidierung, Verschlüsselung)

Nachteile:

- fast keine Unterstützung für Multimediafunktionen (z.B. Audio- und Videoconferencing), Application Sharing oder Whiteboard

- kein richtiges, für alle nutzbares und gestaltbares Dokumentenablagesystem

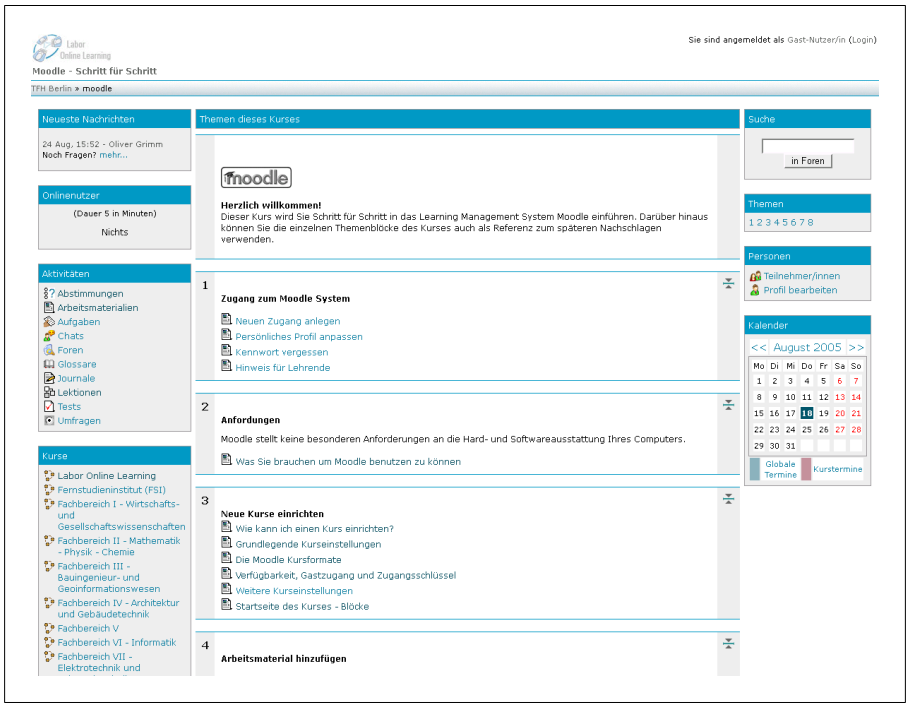


Abb. 4.8: Moodle, Startseite [moo]

4.2.4 Überblick der evaluierten LMS

In Tab. 4.1 auf der folgenden Seite erfolgt ein Überblick aktueller LMS, die zumindest eine teilweise SCORM-Unterstützung implementiert haben. Obwohl SCORM 2004 nun schon vor über einem Jahr veröffentlicht wurde, sind nur eine Hand voll LMS zertifiziert worden.

Produkt	Version	Firma	Lizenz	Implementierte Standards
ATutor	1.5.1		Open Source	IMS, SCORM Content Packaging
WebMentor(*)	4.2	Avilar Technologies, Inc.	kommerziell	AICC AGR-010, SCORM 1.2
Bazaar	7		Open Source	IEEE, IMS
Blackboard Learning System(*)	6 ML	Blackboard, Inc.	kommerziell	IMS Metadata 1.2, IMS CP 1.1.2, SCORM 1.2, Microsoft LRN 3.0
Claroline	1.5		Open Source	IMS 1.1.2, SCORM 1.2 (minimal conformance level + optional Data Model Element) und SCORM 2004 (basic conformance), LDAP
Clix(*)	4.5	IMC GmbH	kommerziell	AICC CMI Level 1, SCORM 1.2
Connected Learning LMS		Connected Learning.Network, Inc.	kommerziell	AICC, IMS, SCORM
Desire2learn(*)(***)		Desire2Learn Inc.	kommerziell	SCORM 1.2, SCORM 2004, IMS Enterprise 1.1, IMS QTI 1.2, IMS Content Packaging Specification
Globalteach		TWI AG	kommerziell	AICC HACP, AICC 3.5, SCORM (API) 1.2
eLearning Suite(***)	2.1	Hyperwave AG	kommerziell	AICC, SCORM 1.2
ForceTen(***)(****)	2.8	Eedo Knowledgeware	kommerziell	SCORM 1.2, IMS Metadata 1.2.1, AICC, IMS Content Packaging 1.1.2
iLearning(***)		Oracle Corporation	kommerziell	SCORM, AICC, IMS
ILIAS	3.4.4	Universität Köln	Open Source	AICC. IMS, Dublin Core Metadata, ARIADNE Educational Metadata Recommendation 3.0, SCORM 1.2
IBM Lotus LMS(***)	1.05	IBM Lotus Software Group	kommerziell	AICC, SCORM 1.2
Moodle	1.5.2		Open Source	SCORM Module
Saba Learning Enterprise(***)	3.4	Saba Software	kommerziell	AICC, IMS, SCORM 1.2
TopClass(***)	6.2	WBT Systems	kommerziell	AICC AGR-010, SCORM 1.2
trainer42		bureau42 GmbH	kommerziell	SCORM, IMS, AICC
WebCT(*)	Vista 3.0	WebCT	kommerziell	SCORM 1.2, MS Content Packaging 1.1.2, IMS QTI 1.1, IMS Enterprise 1.1, and Microsoft LRN 2.0, OKI
.LRN	2.0.3		Open Source	IMS

Tab. 4.1: LMS und unterstützte Standards; (*) SCORM 1.2 spezifiziert, (**) SCORM 2004 spezifiert, SCORM 1.2 Adopter (***), SCORM 2004 Adopter (****)

4.2.5 Standalone Applikationen

Während bei den zuvor vorgestellten LMS (Learning-Management-System) SCORM als Modul integriert ist bzw. als Zusatzfeature angeboten wird, gibt es auch Stand-alone-Applikationen, welche sich nur auf die Erstellung von SCORM-Inhalten bzw. das Ausführen von SCORM-Kursen konzentrieren. Im folgenden werden die bekanntesten Applikation kurz vorgestellt:

RELOAD Editor: Der RELOAD Metadata and Content Package Editor[22] (siehe Abb. 4.9) ermöglicht Autoren das Erstellen und Editieren von IMS- und SCORM (1.2 und 1.3)-konformen Content Packages. Der integrierte Metadaten Editor unterstützt die Beschlagwortung von Lernobjekten nach dem LOM Standard.

Die Software wurde im Rahmen eines vom JISC (Joint Information Systems Committee)[23] geförderten Projektes entwickelt mit dem Ziel, vollständig kompatible Lerntechnologien anzuwenden und umzusetzen. Die Software ist ein in Java geschriebenes Open Source Projekt und damit frei verfügbar.

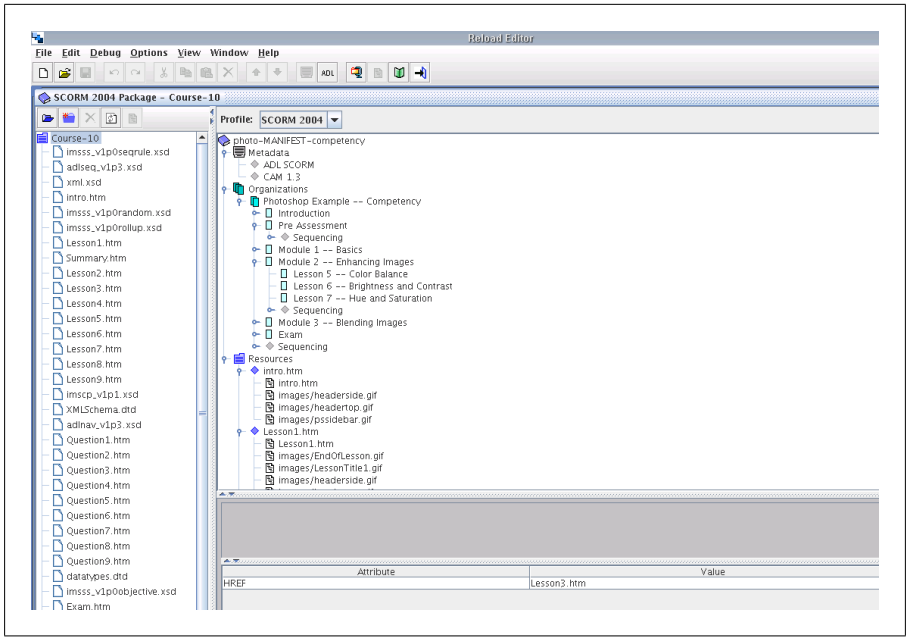


Abb. 4.9: RELOAD Editor 2004

[22] http://www.reload.ac.uk/editor.html
[23] http://www.jisc.ac.uk

RELOAD Player: Die aktuelle Version des RELOAD Player[24] unterstützt ADL SCORM 1.2 und ermöglicht das Betrachten von SCORM Content Packages. Wie auch der RELOAD Editor ist der SCORM Player in Java geschrieben und Open Source.

THESIS[25]: Dieses Tool ermöglicht es, aus Microsoft Word-, Powerpoint-, Flash- oder HTML-Dokumenten, SCORM 1.2-konforme Inhalte zu erzeugen, die in SCORM konformen Learning-Management-Systemen eingesetzt werden können. Das Benutzeroberflächendesign von THESIS ist in MS Office Word und PowerPoint integriert.

Manifest Maker[26]: Dies ist eine kostenlose Erweiterung für Macromedia Dreamweaver MX zur Generierung eines SCORM 1.2 Manifests, das zumindest den Anforderungen für eine SCORM-Konformität entspricht.

SCORM Visualizer[27]: Diese kommerzielle Software stellt eine SCORM 1.2 Laufzeitumgebung für einen E-Learning-Kurs zur Verfügung. Dies ermöglicht Kursentwickler zu prüfen, ob sich der Kurs in einem LMS (Learning-Management-System) starten lässt. Weiters wird ein Kurs bis zum Zertifizierungslevel SCORM LMS RTE3 unterstützt.

CopperCore[28]: In der aktuellen Version 2.2 ist CopperCore eine J2EE Laufzeitumgebung für das IMS Learning Design. Zielgruppe sind Systementwickler, die das IMS Learning Design in eigene Applikationen integrieren möchten. CopperCore unterstützt mit der neuesten Version IMS Learning Design Level A, B und C. Die Software ist Open Source und unterliegt der GPL (GNU General Public License).

[24] http://www.reload.ac.uk/scormplayer.html
[25] https://www.hunterstone.com/hsstore/Products.aspx
[26] http://www.e-learningconsulting.com/products/manifest-maker.html
[27] http://www.e-learningconsulting.com/products/scorm-visualizer.html
[28] http://coppercore.org

4.3 Abschließende Betrachtungen

Zwischenzeitlich sind bereits sehr viele Lern-Management-Systeme, sowohl kommerziell-proprietäre als auch Open Source, verfügbar. Wie Evaluierungen belegen, können Open Source Produkte teilweise in Funktionsumfang und Produktreife durchaus mit kommerziellen System mithalten.

Generell weisen LMS beider Varianten in vielen Bereichen noch etliche Entwicklungsdefizite auf und sind hinsichtlich der didaktischen Perspektive noch stark verbesserungswürdig. Es reicht nicht aus, durch eine Fülle an Features alle denkbaren Anforderungen zu erfüllen. Vielmehr ist es wichtig, ein Produkt nicht zu komplex werden zu lassen und sich auf jene Funktionen zu beschränken, die von einem möglichst großen Anwenderkreis genutzt werden können.

Eine Erfüllung spezifischer Kundenanforderungen ist für kommerzielle Produkte oftmals schwierig, da zwar Feedback und Anregungen der Anwender erwünscht und wichtig sind, die Produkte aber an die Release-Planung des Herstellers gebunden sind. Open Source-Produkte erlauben es, Anwendern durch ihre Lizenzbestimmungen, eigenständig die Software an bestimmte Anforderungen anzupassen und weiterzuentwickeln. Dies ist jedoch relativ anspruchsvoll, weil dafür spezielle Kompetenzen erforderlich sind. Didaktische Anforderungen müssen in technische Spezifikationen übersetzt werden, die dann in der Software realisiert werden sollen. Weiters müssen sich Programmierer von Open Source-Projekten neben Kompetenzen zur kollaborativen Entwicklung der Software auch technische Kompetenzen im Umgang mit speziellen Entwicklertools aneignen. Anders als in Hochschulen sind in Unternehmen solche speziellen Kompetenzen oftmals nicht vorhanden, da dafür weder das nötige Personal noch die Zeit zur Verfügung stehen.

Ein weiteres Entscheidungskriterium für einen Plattformtyp stellen die oftmals hohen Lizenzkosten kommerzieller Produkte dar. Ob diese immer gerechtfertigt sind, sei dahingestellt, Open Source-Systeme scheinen nur im ersten Moment aufgrund fehlender Lizenzierungskosten billiger zu sein. Für beide Systeme fallen zusätzliche Ausgaben für groß angelegte Serverinstallationen an und Mittel für Integration, Anpassung, Wartung uns Sicherheit sind auch aufzubringen. Diese Kosten sind in der Regel höher als die reinen Lizenzkosten.

Zusammenfassend lässt sich feststellen, dass es unabdingbar ist, im Einzelfall zu prüften, für welches Einsatzszenario welches Produkt, sei es nun kommerziell oder Open Source, zur Verfügung steht und die jeweiligen Anforderungen im gewünschtem Maß erfüllt.

Literatur: [HW04], [Fes03], [Koc02], [Exn05], [GEGB04], [RKMH05], [Per05], [Gel05], [HK03], [sof], [moo], [est], [min], [ili05], [Lea04], [Wik], [MHH02], [ete], [LS04b], [MG04], [TES04], [Lea05], [ALB04],[imc04], [HB03], [Edu], [uni], [Sch05]

Teil II

Gestaltungsbereich

Kapitel 5

Das Struts Framework

In diesem Kapitel erfolgt eine überblicksmäßige Einführung in das Struts Framework. Neben einer Beschreibung der Architektur und einzelner Komponenten werden auch sinnvolle Modifikationen und Erweiterungen hinsichtlich des Frameworks vorgestellt. Struts bildet samt den Erweiterungen die Präsentationsschicht der in Kapitel 6 auf Seite 135 vorgestellten Web-Applikation und somit einen elementaren Teil der Anwendung.

Auf eine explizite Beschreibung der Installation und Konfiguration wird hier bewusst verzichtet, eine genaue Erläuterung dahingehend ist in Anhang B zu finden.

5.1 Das MVC-Paradigma

Die grundsätzliche Idee von Model-View-Controller besteht in einer Trennung der Objekte, welche die Logik einer Applikation und die Daten repräsentieren, von der konkreten Darstellung und Manipulation der Daten. Dieser Ansatz ist vor allem dann vorteilhaft, wenn man komplexe Anwendungen in kleinere, logischere Einheiten zerlegen will. Das MVC-Konzept wurde erstmals bei Smalltalk-80 umgesetzt. Es umfasst drei wesentliche Elemente:

- Model (Datenmodell)
- View (Präsentation)
- Controller (Ablaufsteuerung)

Das Model repräsentiert das Datenmodell einer Anwendung und deren aktuellen Zustand. Es ist unabhängig von einer bestimmten Darstellung der Ausgabe oder vom Verhalten der Eingabe. Änderungen werden registriert und der View-Komponente mitgeteilt.

Die View entspricht der Präsentationsschicht einer Model-Komponente. Es können alle Informationen oder nur Teile des Models dargestellt werden. Sie sendet das Benutzerverhalten an den Controller oder es wird der aktuelle Zustand des Models abgefragt.

Der Controller, die Ablaufsteuerung der Anwendung, bestimmt das Verhalten, mit welchem auf Benutzereingaben mittels einer Benutzerschnittstelle reagiert wird. Es werden dabei Dienste der entsprechenden View oder des Models aufgerufen – jeder Controller ist dabei genau einer View zugeordnet, mehrere Controller einem Model. Mit dem Controller wird der Zustand des Models verändert, weiters können Eigenschaften der Ausgabe manipuliert werden.

Die Vorteile des Model-View-Controller Paradigmas liegen in seiner hohen Flexibilität, es sind Änderungen an Teilobjekten und verschiedene Views möglich. Weiters ist die Aufteilung in kleinere Objekte übersichtlicher und einfacher wartbar, auch wird dadurch eine Wiederverwendbarkeit gefördert. Durch die klare Trennung in drei Schichten (siehe Abb. 5.1) wird die Ablaufsteuerung einer Anwendung merklich vereinfacht.

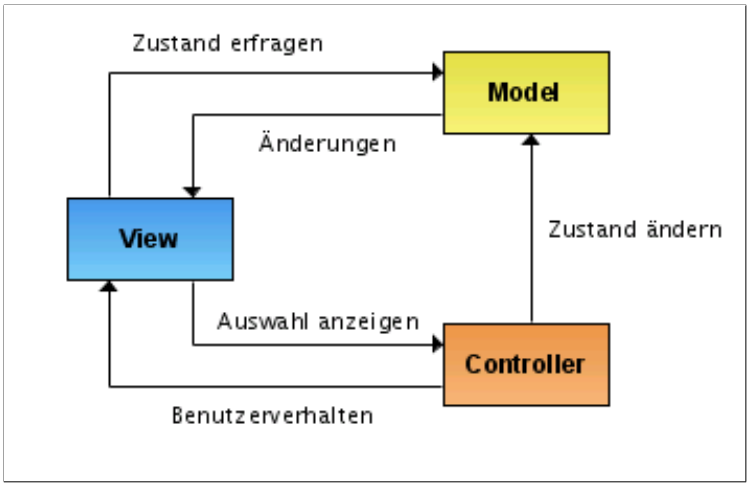


Abb. 5.1: Model-View-Controller Prinzip

5.1.1 Model 1 Architektur

Nach Einführung von Java-Servlets wurde bald erkannt, dass sich eine Erstellung von HTML-Seiten mit Hilfe von Servlets als sehr zeitaufwendig gestaltet. Mittels Java Server Pages versuchte man dies zu umgehen. Viele Web-Applikationen wurden zunächst zur Gänze mittels JSP realisiert. Innerhalb voneinander großteils unabhängiger JSPs wurden Model, View und Controller implementiert. Dieser Ansatz wird auch als „Model 1 Architektur“ bezeichnet.

Für kleinere Applikationen eignet sich dieser Architektur durchaus, erst bei komplexeren Strukturen kommt es vermehrt zu Nachteilen. Ein Vermischen verschiedener Sprachen und das Zusammenführen von Geschäftslogik, Ansicht und Ablaufsteuerung bilden einen äußerst unübersichtlichen Code, der nur schwer wartbar ist. Somit gibt es auch keine klare Trennung zwischen Programmierer und Webdesigner - eine Aufteilung der Entwicklungsarbeit ist nur schwer möglich, ebenso eine Wiederverwendbarkeit des Codes. Änderungen in der Codestruktur lassen sich aufgrund fehlender Modularisierung nicht zentral vornehmen, sondern müssen an vielen verschiedenen Stellen durchgeführt werden.

5.1.2 Model 2 Architektur

Betrachtet man die Nachteile der Model 1 Architektur, war die logische Konsequenz eine Trennung von Web-Design und Programmierlogik. Diese Idee wurde im bereits erwähnten MVC Paradigma von Smalltalk erstmals verwirklicht.

Für eine Verwendung innerhalb einer Web-Applikation musste das klassische MVC jedoch noch angepasst werden. Hier ist die Verbindung zwischen Client und Server statuslos, wodurch es dem Model nur schwer möglich ist, die Präsentationsebene über Datenänderungen zu informieren. Ein Web-Browser muss den Server immer wieder abfragen, um so Änderungen an den Daten oder dem Status einer Anwendung zu registrieren. Weiters wird nur mehr für die Präsentationsschicht JSP verwendet, Model und Controller sind in Java implementiert (siehe Abb. 5.2 auf der folgenden Seite). Diese Abwandlung vom klassischen MVC wird als „Model 2“, oftmals auch als „MVC 2“ oder „MVC Model 2“ bezeichnet.

5.2 Das Framework

Struts[1] ist ein leistungsfähiges Open Source Framework, das versucht, die Vorteile von Java Server Pages und Servlets zu vereinen. Die erste Version wurde von der Apache Software Foundation[2] Anfang 2001 veröffentlicht. Das Framework entspricht einer MVC Model 2 Implementierung und setzt sich somit aus Model, View und Controller zusammen (siehe Abb. 5.2). Zentraler Bestandteil des Frameworks stellt die Konfigurationsdatei `struts-config.xml` dar.

Im folgenden werden die wichtigsten Komponenten von Struts eingehender betrachtet.

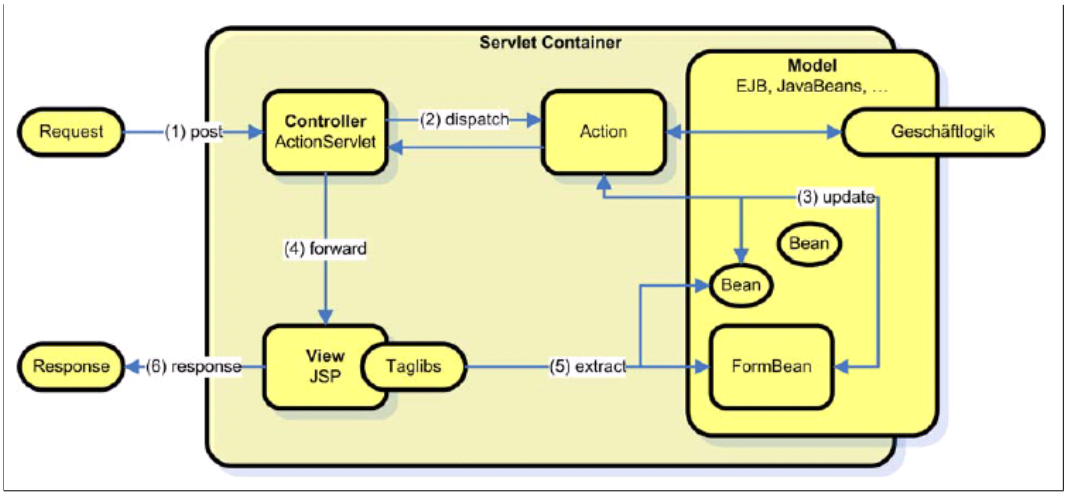


Abb. 5.2: Model 2 Architektur des Struts Framework [Ack04]

5.2.1 Model Komponenten

Das Model kann in zwei Bereiche unterteilt werden: Einerseits repräsentiert es den aktuellen Zustand einer Applikation, andererseits kapselt es die Geschäftslogik ab. Für beide Fälle werden im Framework so genannte *Java-Beans* eingesetzt.

[1] http://struts.apache.org
[2] http://apache.org

5.2.1.1 Beans für die Geschäftslogik

In diesem Zusammenhang spricht man auch von *Business Logic Beans*. Der Aufruf erfolgt über Action-Klassen (siehe Abschnitt 5.2.3.3 auf Seite 120). Beans bestehen aus verschiedenen Methoden, in welchen die Geschäftslogik zur Anwendung kommt. Dies hat den Vorteil, dass die Logik von den Daten getrennt ist. Business Logic Beans können somit nicht nur innerhalb einer bestimmten Web-Applikation eingesetzt werden, sondern können auch in anderen Anwendungen zur Ausführung der Geschäftslogik verwendet werden – vorausgesetzt, Java-Beans werden so entworfen und implementiert, dass sie in Unkenntnis sind, in welche Web-Anwendung sie ausgeführt werden. Es sollten daher keine Import-Anweisungen für Java-Klassen implementiert sein. Eine Action-Klasse soll stattdessen alle notwendigen Informationen aus dem HTTP-Request extrahieren und den Beans als Methodenparameter oder mittels Setter-/Getter-Methoden übergeben. Je nach Applikationskomplexität handelt es sich bei den Beans um gewöhnliche Java Beans, in größere Anwendungen kommen häufig *Enterprise Java Beans* zum Einsatz.

5.2.1.2 Beans für den Systemzustand

Der aktuelle Zustand einer Applikation wird durch eine oder mehrerer Java-Beans repräsentiert, man spricht auch von *System State Beans*. Ein Anwendungsbeispiel hierfür wäre ein Warenkorb innerhalb eines Web-Shops, der mittels eines Beans realisiert wird.

In kleinen Web-Anwendungen werden System-State-Beans für das Speichern von Zustandsinformationen, die nicht längere Zeit gespeichert werden müssen, verwendet. In den meisten Fällen erhalten Beans Daten, die permanent in einer externen Datenbank gespeichert werden – sie werden bei Bedarf erzeugt, danach wieder aus dem Speicher des Servers entfernt. Für größere Applikationen werden auch so genannte *Entity Enterprise Java Beans* eingesetzt.

5.2.1.3 ActionForm-Beans

In der Präsentationsschicht werden ActionForm-Beans für ein Zwischenspeichern und Validieren von Formulardaten in HTML-Seiten bzw. JSPs verwendet. Prinzipiell ist jedem Formular ein ActionForm-Bean zugeordnet, sofern es in der Struts-Konfigurationsdatei definiert ist. ActionForm-Beans sind von der Klasse `ActionForm` abgeleitet. Sie sind Java-Beans, deren Attribute mit den Feldnamen in einem HTML-Formular korrespondieren. Der Zugriff auf die Attribute erfolgt mittels entsprechender Setter- und Getter-Methoden.

ActionForm-Beans bieten auch die Möglichkeit einer automatischen Validierung. Hierfür muss die Methode `validate()` überschrieben werden. Innerhalb dieser Methode können Formularwerte nach bestimmten Regeln überprüft werden. Treten bei der Validierung Fehler auf, besteht die Möglichkeit, entsprechende Fehlermeldungen innerhalb der jeweiligen Formularseite auszugeben.

Es ist nicht immer sinnvoll, die Validierung in ActionForm-Beans vorzunehmen. Soll ein Java-Bean für mehrere Formulare verwendet werden, erweist es sich als sinnvoller, die Überprüfung in der Action-Klasse durchzuführen – als Beispiel seien Wizard-Anwendungen genannt.

Anstelle der Validierung mittels ActionForms kann auch ein eigener Validierungsmechanismus via Plugin implementiert werden, mehr dazu in Abschnitt 5.3.1 auf Seite 122.

5.2.2 View Komponenten

Die Präsentationsschicht von Struts wird für die Darstellung der Daten verwendet. Die Realisierung erfolgt üblicherweise mittels Java Server Pages. Die Seiten können sowohl statisches HTML- oder XML enthalten, als auch zur Laufzeit dynamisch generierten Code.

Struts beinhaltet auch umfassende Tag-Bibliotheken (*Tag-Libraries*), die es ermöglichen, Views zu generieren, welche vollständig internationalisierbar sind und mit ActionForm-Beans interagieren können. Weiter Vorteil ist, dass dadurch vermehrt Java-Code aus der View verbannt werden kann. Eine Einbindung der Biblio-

theken erfolgt zu Beginn des jeweiligen JSPs.

5.2.2.1 Struts-Tag-Libraries

Tag-Libraries, kurz „Taglibs“, sind ein wesentlicher Bestandteil des Frameworks. Unter Tags versteht man Formatierungselemente in HTML- oder JSP-Seiten, die vom Client zur Darstellung von Daten benutzt werden. Weiters existieren Tags, die im Server zur einer logischen Formatierung der Daten benutzt und in HTML-Tags transformiert werden. Bei Struts unterscheidet man zwischen folgenden Libraries:

- `html`: Stellt eine Erweiterung der HTML Tags dar, insbesondere für Formularelemente.
- `bean`: Ermöglicht den Zugriff auf Java-Beans und deren Eigenschaften.
- `logic`: Stellt logische Komponenten wie Iteration oder auch konditionale Elemente zur Verfügung.
- `tiles`: Ist eine mächtige Template-Erweiterung und ermöglicht es Entwicklern, Webseiten komponentenbasiert aufzubauen.

Bei der Anwendung von Struts üblichen Tag-Bibliotheken besteht die Möglichkeit, alternative Bibliotheken zu verwenden. Im Moment ist eine Ablösung der Logic- und Bean-Tags durch JSTL (Java Server Pages Standard Tag Library) (siehe Kapitel 5.3.5.1 auf Seite 127) in Arbeit. Weiters sollen HTML-Tags durch Expression Language-Tags (siehe Kapitel 5.3.5.2 auf Seite 128) ersetzt werden.

5.2.2.2 Internationalisierung

Ein nützliches Feature von Struts bietet die Möglichkeit einer Internationalisierung von Web-Applikationen. Hierfür müssen entsprechende Properties-Dateien (`ApplicationResources.properties`) im Klassenpfad der Applikation verfügbar sein – die Definition erfolgt innerhalb der Struts-Konfigurationsdatei (siehe Listing 5.1 auf Seite 121, Zeile 19). Für jede unterstützte Sprache muss neben der originalen Properties-Datei eine entsprechende weitere Datei angelegt werden. Mittels Message-Tags (siehe Listing 5.7 auf Seite 128, Zeile 2) sind in der jeweiligen View nur mehr Verweise auf enthaltene Texte in der entsprechenden Properties-Datei an-

zugeben. Texte müssen somit nicht mehr „hardcoded“ in ein JSP-File geschrieben werden.

5.2.3 Controller

Der Struts Controller stellt das zentrale Element innerhalb des Frameworks dar und ist für den gesamten Ablauf einer Struts-Anwendung zuständig. Seine Aufgabe besteht darin, Requests vom Client entgegen zunehmen, dabei zu entscheiden, welche Funktion der Geschäftslogik auszuführen ist und als Response eine entsprechende View-Komponente (JSP) auszuwählen. Der Vorteil des Controllers liegt in der zentralen Überwachung und Konfiguration.

Der Controller umfasst folgende Komponenten:

- Klasse `ActionServlet`
- Klasse `ActionMapping`
- Konfigurationsdatei `struts-config.xml`
- weitere Klassen, die sich von der Klasse `Action` ableiten

5.2.3.1 ActionServlet

Das Servlet stellt die primäre Komponente in der Ablaufsteuerung dar. Es erzeugt und benutzt Action-Objekte für die Ausführung spezieller Aufgaben, ActionForm-Beans für das Speichern und Validieren von Formulardaten und ActionForward-Klassen für eine Weiterleitung des Programmflusses.

Eine Anfrage an die Web-Anwendung entspricht meistens einem HTTP-Request, der vom Browser in Form eines Request-URI übermittelt wird. Der Controller liest diese aus und entscheidet, welche Action-Klasse aufgerufen werden soll. Die entsprechende Klasse wird via dem ActionMapping ermittelt.

5.2.3.2 ActionMapping

Die Klasse ActionMapping bildet also einen eingehenden Request-URI (Ereignis) auf die zuständige Action-Klasse ab. ActionMappings werden in der zentralen Struts-Konfigurationsdatei (`struts-config.xml`) definiert, wodurch es einfach ist, eine Ablaufänderung durchzuführen, ohne Änderungen im Code vornehmen zu müssen.

In Listing 5.1 auf der folgenden Seite wird ein Teil einer typischen Konfigurationsdatei dargestellt: Innerhalb der `<action-mappings>`-Tags erfolgt die Einbindung beliebig vieler Aktionen. Jedes Ereignis (Aktion) verfügt über ein eigenes `<action>`-Element, das folgende Attribute beinhaltet:

path beschreibt den relativen Pfad eines Ereignisses.

type definiert den vollständigen Klassennamen der Action-Klasse.

name bezeichnet den Namen des FormBean.

scope definiert den Gültigkeitsbereich in dem ein Action-Bean existiert.

input ist jenes JSP, von dem aus der Request abgeschickt wurde und im Fehlerfall zurückgesendet wird.

Die `<forward>`-Elemente legen fest, wie die Rückgabewerte der Action-Klasse interpretiert werden sollen. Ein Forward kann ein JSP oder wieder eine Action sein. Der Bereich `<globalforwards>` enthält Forward-Elemente, die JSPs entsprechende logische Namen zuordnen.

5.2.3.3 Action

Für jede logische Abfrage wird eine Action-Klasse zur Verfügung gestellt. Diese verarbeitet den Request und gibt ein ActionForward-Objekt zurück, das diejenige View definiert, die den Response generiert. Die Klasse dient als eine Wrapper-Klasse für die Geschäftslogik, wodurch eine Web-Anwendung modular gehalten werden kann.

Die Action-Klasse übersetzt einen HttpServletRequest und ruft innerhalb der

```
1  <struts-config>
2
3   <action-mappings>
4      <action path="/message"
5        type="edu.iicm.messagebaord.message.MessageAction"
6        name="messageForm"
7        scope="request"
8        input="/player/message.jsp"
9        validate="false">
10       <forward name="new" path="/player/message.jsp"/>
11       <forward name="success" path="/player/forward.jsp"/>
12       <forward name="show" path="/player/showMessage.jsp"/>
13       <forward name="related" path="/player/showThread.jsp"/>
14       <forward name="failure" path="/player/message.jsp"/>
15       <forward name="cancel" path="/player/forward.jsp"/>
16     </action>
17  </action-mappings>
18
19  <message-resources null="false" parameter="ApplicationResources"/>
20
21  <plug-in className="org.apache.struts.validator.ValidatorPlugIn">
22    <set-property property="pathnames"
23      value="/WEB-INF/validator-rules.xml,/WEB-INF/validation.xml"/>
24  </plug-in>
25
26 </struts-config>
```

Listing 5.1: Beispiel für ein ActionMapping (`struts-config.xml`)

`execute()`-Methode (Listing 5.2 auf der nächsten Seite, Zeile 2) die Geschäftslogik auf. Weiters wird der Methode auch die ActionForm (siehe Listing 5.2 auf der folgenden Seite, Zeile 3) übergeben, die durch ein FormBean repräsentiert wird und Daten der View enthält. Die Daten werden dann ausgelesen und die entsprechende Geschäftslogik ausgeführt. Zurückgegeben wird ein ActionForward-Objekt (siehe Listing 5.2 auf der nächsten Seite, Zeile 7), das die nächste aufzurufende Seite bestimmt (siehe Abschnitt 5.2.3.2 auf der vorhergehenden Seite).

```
1 public class MessageAction extends Action {
2   public ActionForward execute(ActionMapping mapping,
3                                ActionForm form,
4                                HttpServletRequest request,
5                                HttpServletResponse response) {
6     ...
7     return (mapping.findForward("success"));
8   }
9 }
```

Listing 5.2: Beispiel für eine Action-Klasse (`MessageAction.java`)

5.3 Modifikationen und Erweiterungen

Im folgenden werden Modifikationen und Erweiterungen hinsichtlich des Struts Frameworks erläutert, welche bei dem im Kapitel 6 auf Seite 135 vorgestellten LMS-Prototypen verwendet und implementiert wurden.

5.3.1 Struts Validator

Struts bietet ein zusätzlich einsetzbares Validierungs-Framework[3], das eine serverseitige und optional eine clientseitige Auswertung mittels JavaScript unterstützt. Der Validator erlaubt eine deklarative Validierung außerhalb von Java, die Konfiguration erfolgt in einer oder mehreren XML-Dateien.

Das Framework verfügt über eine Reihe an vordefinierten Validatoren, die eine einfache Überprüfung von Formularen in der View ermöglichen. Beispiele hierfür sind: `required`,`minlength`,`maxlength`, `date`. Seit Struts in der Version 1.2 ist zusätzlich das Jakarta Common Validator-Projekt[4] eingebunden.

Die Validierung von Formulareingaben erfolgt, indem an entsprechender Stelle in der View mittels Error-Handling eine fehlerspezifische, festgelegte und internationalisierte Fehlermeldung platziert wird. Der Validator geht sequentiell alle Felder in dessen Konfigurationsdatei (siehe Listing 5.3 auf Seite 124) durch, holt sich

[3] http://struts.apache.org/userGuide/dev_validator.html
[4] http://jakarta.apache.org/commons/validator/

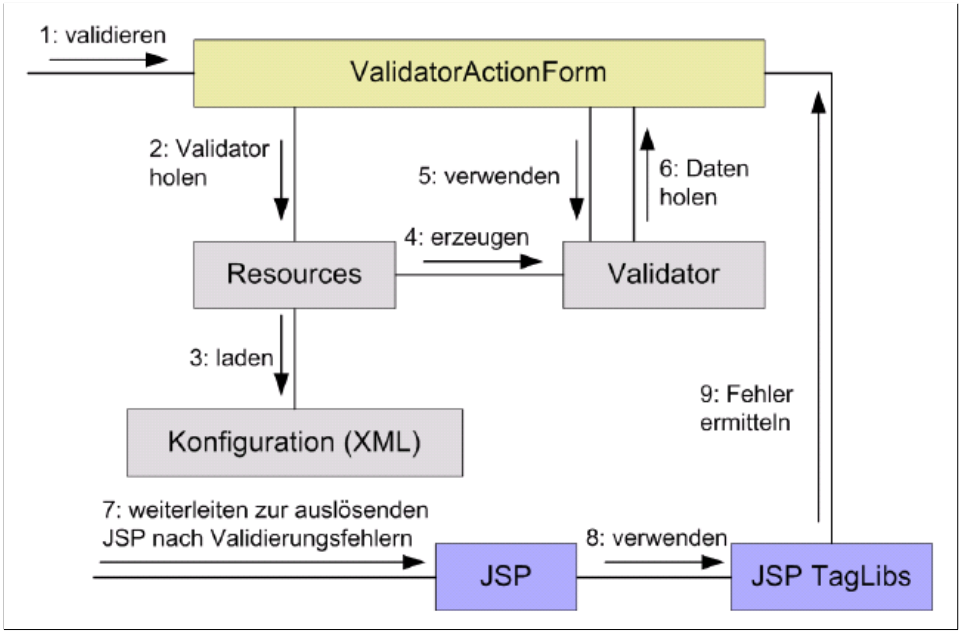


Abb. 5.3: Das Struts Validator Framework [Bee03]

für jedes Feld die Daten aus dem zugehörigen FormBean und validiert sie entsprechend der Konfiguration.

Die Einbindung in das Struts Framework erfolgt zentral über die Plugin-Schnittstelle innerhalb der Struts-Konfigurationsdatei (siehe Listing 5.1 auf Seite 121, Zeile 21 bis Zeile 24). Der Validator wird stets vor eine Action ausgeführt. Um das Framework zu nutzen, muss die FormBean von der Klasse `ValidatorForm` abgeleitet sein und die Methode `validate()` ausführen.

Die Vorteile des Frameworks liegen einerseits in der Möglichkeit der clientseitigen Auswertung und andererseits in der leichten Anpassbarkeit und Erweiterbarkeit. Mit den standardmäßig verfügbaren Validatoren werden bereits häufig zu validierende Bereiche abgedeckt. Sollten diese dennoch nicht den Anforderungen entsprechen, ist eine Erweiterung durch eigene Validatoren möglich.

5.3.2 DynaActionForms

Bei Projekten größeren Umfangs kann es mitunter zeitaufwändig und vor allem sehr unflexibel sein, ausschließlich FormBeans zu verwenden. Eine Möglichkeit

```
1  <form name="messageForm">
2    <field property="message" depends="required">
3      <arg0 key="error.message.required"/>
4    </field>
5    <field property="subject" depends="required">
6      <arg0 key="error.subject.required"/>
7    </field>
8  </form>
```

Listing 5.3: Beispieleintrag in der Konfigurationsdatei des Validators (`validation.xml`)

diesen Unzulänglichkeiten auszuweichen, sind so genannte *DynaActionForms*. Es handelt sich dabei um generische FormBeans, welche vom Struts Framework dynamisch mit allen Request-Parametern befüllt werden, die im entsprechenden Formular enthalten sind. Der große Vorteil besteht nun darin, dass auf eine Implementierung expliziter Klassen verzichtet und dadurch auch einfacher und flexibler auf Änderungen in den View-Komponenten reagiert werden kann.

Die Definition erfolgt deklarativ im Konfigurations-File (`struts-config.xml`, siehe Listing 5.4): Für jedes Element des Formulars muss ein `<form-property>`-Element (Zeile 3) definiert werden. Das Attribut `name` (Zeile 3) definiert den Namen der jeweiligen Property. Das Attribut `type` (Zeile 3) teilt dem ActionServlet mit, um was für einen Typ es sich bei dieser Property handelt. Wenn Struts nun auf das FormBean `messageForm` zugreifen will, instanziert es dynamisch ein JavaBean mit einer get- und einer set-Methode für jede definierte `<form-property>`. Die Request-Parameter werden dabei in einer Hashmap abgelegt - via entsprechender Methoden der Form `get(parameterName)` kann darauf zugegriffen bzw. Werte mit entsprechender set-Methode gesetzt werden. Auch die Methode `reset()` wird automatisch generiert. Struts löscht bei deren Aufruf einfach die Inhalte aller Properties.

```
<form-bean name="messageForm"
           type="org.apache.struts.validator.DynaValidatorForm">
  <form-property name="id" type="java.lang.String" initial=""/>
  <form-property name="action" type="java.lang.String" initial=""/>
  <form-property name="subject" type="java.lang.String" initial=""/>
  <form-property name="body" type="java.lang.String" initial=""/>
  <form-property name="type" type="java.lang.String" initial=""/>
  <form-property name="types" type="java.util.ArrayList"/>
</form-bean>
```

Listing 5.4: Definintion einer DynaActionForm innerhalb der Struts-Konfigurationsdatei

DynaValidationForms

Eine weitere Variante für die Implementierung einer ActionForm und Erweiterung der DynaActionForms stellen so genannte *DynaValidationForms* dar. Zusätzlich zu den zuvor beschrieben Eigenschaften erfolgt hier die Validierung nicht mehr von Struts selbst, sondern vom Apache Validator Framework. Die `validate()` Methode ist zwar vorhanden, ist aber leer und hat keinerlei Funktionalität.

Neben DynaValidationForms gibt es noch *DynaValidationActionForms*. Der Unterschied besteht dabei im ActionMapping: Bei DynaValidationForms entspricht das Mapping dem Namen des FormBean, bei DynaValidationActionForms wird der Pfad der Action gemappt.

5.3.3 Dynamic Transfer Object

Beim Versenden von Daten über Netzwerke mit einem einzigen Transfer muss eine Vielzahl an Parametern übertragen werden. Nachdem Java nur maximal einen Rückgabewert haben darf, wäre für jede Information ein eigener Aufruf notwendig. Dadurch kann es unter Umständen zu Performanceeinbußen kommen - Abhilfe schaffen so genannte *Dynamic Transfer Objects*, oftmals spricht man auch von *Value-Objects*. Darunter versteht man ein Objekt, das alle Informationen kapselt, womit nur ein Parameter übertragen werden muss.

Ein DTO (Dynamic Transfer Object) entspricht prinzipiell einem einfachen Java-

Bean und besitzt Attribute jener Daten, die in einer Client-Anwendung dargestellt oder eingegeben werden. Für die Attribute werden vom DTO getter- und setter-Methoden zur Verfügung gestellt – ein DTO ist also nur ein Datencontainer. Mittels Serialisierung werden die Daten in einem binären Format vom Server zurück zum Client übertragen.

5.3.4 Struts Menu

Struts Menu[5] ist ein von Matt Raible initiiertes Open Source Projekt mit Ziel, auf relativ einfache Weise Menüstrukturen für die Navigation innerhalb einer Struts-Applikation zu integrieren. Das Projekt besteht aus Tag-Libraries die als Plugin in der `struts-config.xml` eingebunden werden.
Die Konfiguration erfolgt mittels einem XML-File (siehe Listing 5.5), in welchem der Menütyp (Zeile 4) und Einträge des jeweiligen Menüs definiert werden. Ein Menü selbst ist als so genanntes Repository im Seitenkontext innerhalb der View-Komponenten (siehe Listing 5.6 auf der nächsten Seite) verfügbar.

Zusätzlich zur Definition statischer Menüs mittels einer XML-Datei besteht die Möglichkeit, auch dynamische Menüs via Datenbankabfragen und ActionServlets zu generieren.

```
1  <MenuConfig>
2    <Displayers>
3      <Displayer name="CoolMenu"
4        type="net.sf.navigator.displayer.CoolMenuDisplayer"/>
5    <Menus>
6      <Menu name="UserMenuTutor" title="User">
7        <Item title="add User" toolTip="Create a new User"
8              location="user.do" target="main"/>
9        <Item title="show Users" toolTip="Show all existing User"
10             location="userlist.do" target="main"/>
11     </Menu>
12   </Menus>
13 </MenuConfig>
```

Listing 5.5: Beispiel einer Menüdefinition (`menu-config.xml`)

[5] http://struts-menu.sourceforge.net

```
<menu:useMenuDisplayer name="CoolMenu" bundle="null">
  <menu:displayMenu name="UserMenuTutor"/>
  <menu:displayMenu name="CourseMenuTutor"/>
</menu:useMenuDisplayer>
```

Listing 5.6: Aufruf des Menüs innerhalb einer JSP-Datei

5.3.5 Tag Libraries

Neben den standardmäßig zur Verfügung stehenden Struts Tag-Libraries erweist es sich als durchaus sinnvoll, weitere, effizientere Bibliotheken zu verwenden.

5.3.5.1 JSTL

Die JSTL (Java Server Pages Standard Tag Library)[6] ist eine Sammlung von vier Custom-Tag-Libraries, welche für die Erstellung von JSPs hilfreich sind. Der Vorteil liegt dabei in der Wiederverwendbarkeit dieser Komponenten bzw. der sauberen Trennung zwischen Oberflächendesign und Logik; komplexe Geschäftslogiken können hinter einfach zu bedienenden Tags ausgelagert werden. Die Lesbarkeit einer Seite und somit auch die Wartbarkeit wird dadurch verbessert. Custom-Tags wurden erstmals mit der Version JSP 1.1 eingeführt.

Die Version 1.1 verfügt über folgende Bibliotheken:

Funktion	**URI**	**Präfix**	Beispiel
Core	http://java.sun.com/jstl/core	c	`<c:when>`
XML Processing	http://java.sun.com/jstl/xml	x	`<x:forEach>`
I18N Capable Formatting	http://java.sun.com/jstl/fmt	fmt	`<fmt:message key=’’ ’’/>`
Relational DB Access (SQL)	http://java.sun.com/jstl/sql	sql	`<sql:query>`

Tab. 5.1: JSTL Tag-Bibliotheken

[6] http://java.sun.com/products/jsp/jstl/

Core Tag Library: Besteht aus iterativen, konditionalen, URL-spezifischen und allgemeinen Tags.

XML Tag Library: Beinhaltet Tags für den Zugriff auf XML-Dokumente und für XML-Transformationen.

SQL Tag Library: Besteht aus Tags für Datenbankzugriffe und für direkte Datenbankverarbeitung.

Internationalisation Tag Library (I18N): Diese Bibliothek dient zur Formatierung und Internationalisierung.

5.3.5.2 Expression Language

Im Zuge aktueller JSTL-Versionen wurde die so genannte Expression Language (EL) eingeführt. Die Expression Language besteht aus Elementen, die an Skriptelemente interpretierter Skriptsprachen angelehnt sind. Jedes Skriptelement beginnt mit einem Dollar-Zeichen und schließt einen Ausdruck (*Expression*) in geschweiften Klammern ein (siehe Listing 5.7) - Daten können so aus Bean-Komponenten ausgelesen werden. Ab der Version JSP 2.0 ist die Expression Language standardmäßig integriert.

Ab Struts in der Version 1.1 wurden die eigenen Tag-Libraries um Struts-EL-Tag-Libraries erweitert. Jedes Custom-Tag in dieser Bibliothek ist mit einem Tag aus der originalen Bibliothek assoziiert mit dem Unterschied, dass für die Interpretation der Tags die *Expression Evaluation Engine* von JSTL verwendet wird. Auf eine gänzliche Portierung wurde jedoch verzichtet, da ansonsten Tags in redundanter Form vorliegen würden.

```
1 <c:if test="${not empty param.action}">
2   <fmt:message key="message.subject"/>
3 ...
4 </c:if>
```

Listing 5.7: Beispiel für EL-Tags

5.3.6 log4j

log4j[7] wurde für das Tracing von Anwendungsmeldungen in Java entwickelt. Es ist im Rahmen eines Open Source-Projekts entstanden und ist heute Teil des Logging-Projektes der Apache Software Foundation. Nachdem ein Logging von Applikationen normalerweise mit Performanceeinbußen verbunden ist, wurde log4j entsprechend den laufzeitkritischen Anforderungen von (Web-) Applikationen optimiert.

Funktionsweise

Auftretende Fehler und Debug-Meldungen werden nicht auf der Standardausgabe ausgegeben, sondern an das Logging-System übergeben. Gleichzeitig wird eine Einteilung nach Prioritäten, so genannten *Levels* vorgenommen. Die Einteilung der Wichtigkeit einer Nachricht wird vom Programmierer festgelegt, Filterung und Art der Ausgabe können zur Laufzeit konfiguriert werden.

In Tab. 5.2 sind die Prioritätslevel aufgelistet, folgende Reihenfolge ist dabei bindend: `ALL < DEBUG < INFO < WARN < ERROR < FATAL < OFF`.

Level	Beschreibung
`ALL`	alle Meldungen werden ungefiltert ausgegeben
`DEBUG`	allgemeines Debugging
`INFO`	allgemeine Informationen
`WARN`	Auftreten einer unerwarteten Situation
`ERROR`	Fehler (Exception)
`FATAL`	kritischer Fehler, Programmabbruch
`OFF`	Logging ist deaktiviert

Tab. 5.2: log4j Prioritätslevel

Mittels so genannter *Appender* (siehe Tab. 5.3 auf der folgenden Seite) besteht die Möglichkeit, Meldungen eines Loggers auf verschiedenste Art auszugeben; es können mehrere Appender an einen Logger angehängt werden.

Weiters kann das Layout der auszugebenden Nachrichten individuell angepasst und konfiguriert werden. Neben der reinen Nachricht kann mittels Muster

[7] http://logging.apache.org/log4j/

Appender	Beschreibung
`ConsoleAppender`	Ausgabe auf Standardausgabe
`FileAppender`	Schreiben in eine Datei
`RollingFileAppender`	Beginnen einer neuen Datei ab einer gewissen Größe
`DailyRollingFileAppender`	Beginnen einer neuen Datei zu gewissen Zeitpunkten
`SyslogAppender`	Loggen mittels Syslog-Dienst
`NTEventLogAppender`	Schreiben in das Ereignisprotokoll des Windows-Systems
`SMTPAppender`	Schicken von Mail bei gewissen Meldungen

Tab. 5.3: log4j Appender

zusätzlich zum Beispiel Wichtigkeit, Datum, Loggername, Klassenname oder Methodenname ausgegeben werden. Der Name des Loggers entspricht dabei dem Klassennamen, wodurch sich die Ausgaben individuell filtern lassen.

Die soeben genannten Konfigurationsmöglichkeiten werden in einer Konfigurationsdatei vorgenommen. Unter Tomcat besteht die Möglichkeit, log4j mittels einem speziellen Servlet zu initialisieren, das in der Konfigurationsdatei (`web.xml`) definiert wird.

5.3.7 Datenbank-Anbindung

In Struts gibt es die Möglichkeit, innerhalb der Konfigurationsdatei Datenquellen (`<data-sources>`) zu definieren. Weiters bietet das Framework einen einfachen JDBC-Connection-Pool. Im folgenden werden kurz Begriffe, Datenbanken (MySQL) und Struts betreffend, erläutert. Der Autor geht jedoch davon aus, dass die grundlegenden Termini bereits bekannt sind. Die Einbindung einer Datenbank in Struts wird in Anhang B in Listing B.1 auf Seite 173 beschrieben.

5.3.7.1 JDBC

JDBC steht für Java Database Connectivity und definiert eine Java-API, die eine einheitliche Schnittstelle zu relationalen Datenbanken (z.B. SQL) bietet. Sämtliche Interaktionen mit einer Datenbank werden als Objekte repräsentiert; eine offene Datenbankverbindung, auch *Connection* genannt, ist ein Beispiel dafür.

Über Treibermanager (DriverManager) werden Datenbanktreiber geladen, die

einheitliche API-Aufrufe in Befehle übersetzen, die das Datenbanksystem (DBMS) verstehen und ausführen kann. Wird nach der Abfrage der Datenbank eine Ergebnismenge zurückgegeben, wird diese durch den Treiber in eine für Java verständliche Form transformiert und dem Java-Programm zur Verfügung gestellt.

Bei JDBC-Treibern wird zwischen vier verschiedenen Treibertypen unterschieden: Typ 1 und 2 sind nicht netzwerkfähig und nur teilweise in Java implementiert. Typ 3 und 4 sind zur Gänze in Java geschrieben und verwenden ein Netzwerkprotokoll, wodurch entfernte Datenbanken angesprochen werden können. Eine Treiberauswahl erfolgt den Anforderungen der jeweiligen Applikation entsprechend. Ein Treiber stellt dabei die eigentliche Schnittstelle zwischen JDBC und einer Datenbank dar.

Ein Datenbankzugriff erfolgt dabei folgendermaßen:

1. Herstellung einer Verbindung zur Datenbank über den entsprechenden JDBC-Treiber für das verwendete DBMS (Database Management System)
2. Erstellung eines Statement-Objektes (statisches SQL-Kommando)
3. Weitergabe des auszuführenden Statements über das Statement-Objekt an das darunterliegende DBMS
4. Abholung der Ergebnisse über die Ergebnisdatensätze
5. Schließen der Verbindung zur Datenbank

5.3.7.2 Database Connection-Pooling

Die Idee bei *Connection-Pooling* ist, offene Connections in Pools zu verwalten. Ein Aufbauen einer Datenbankverbindung benötigt gewisse Zeit und erfordert eine Bereitstellung entsprechender Ressourcen seitens des DBMS – ein häufigerer Aufbau von Connections bzw. eine größere Anzahl dieser vorzuhalten bringt erhebliche Performanceeinbußen mit sich. Mittels Connection-Pooling wird eine Verbindung aus dem Pool geholt und nur so lange behalten, wie sie benötigt wird. Danach schließt man diese nicht physisch, sondern gibt sie an den Pool zurück. Gerade bei Server-Applikationen, wo viele Client-Anfragen parallel bearbeitet werden müssen, erweist sich dies als sehr sinnvoll.

So genannte *DataSources* bieten einen eleganten Weg zur Benutzung und Verwaltung von Connection-Pooling. In Abb. 5.4 wird der Zusammenhang zwischen Middleware und dem JDBC-Treiber dargestellt: Die Applikation fordert bei einer DataSource eine Connection an, dabei wird eine „PooledConnection“ aus dem Pool geholt. Sofern keine passende Datenbankverbindung vorliegt, wird im Pool eine neue angelegt. Die Datasource verwendet die PooledConnection, um ein Connection-Objekt zu erzeugen, welches das zugängliche Interface für die Applikation implementiert. Nur dieses Objekt wird von der Applikation benutzt. Sollte es nicht mehr verwendet werden, wird die PooledConnection informiert, die wiederum dem Connection Pool mitteilt, dass sie wieder zur Verfügung steht.

Für Connection-Pooling muss also der JDBC-Treiber Implementierungen für die Interfaces `ConnectionPoolDataSource`, `PooledConnection` und `Connection` bereitstellen. Die Middleware implementiert die DataSource. Wie ein Pool implementiert ist, wird im Standard nicht festgelegt und bleibt der Middleware überlassen.

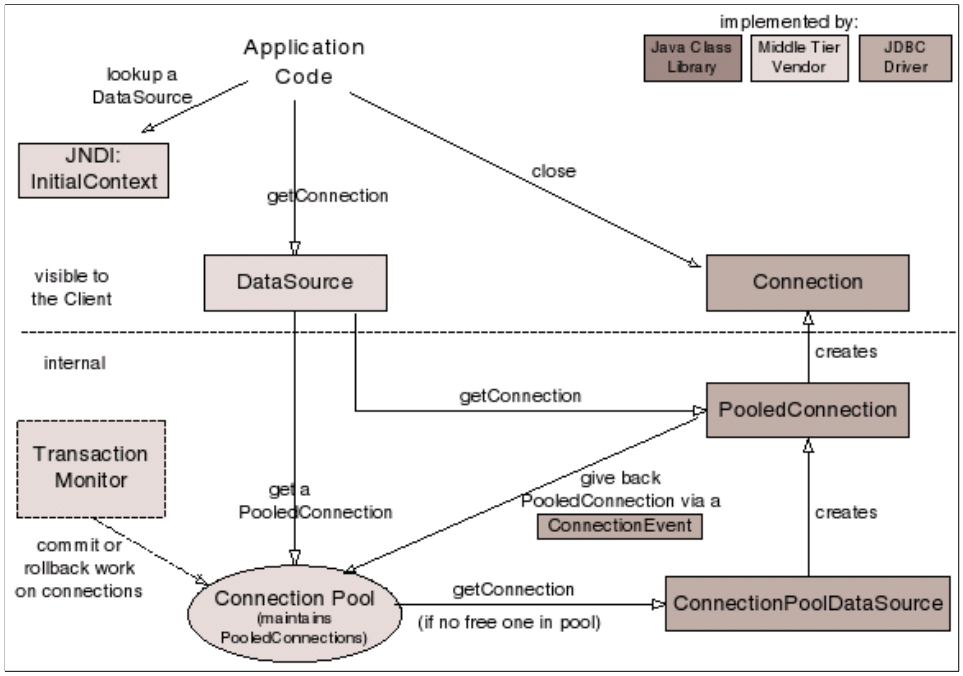


Abb. 5.4: Connection-Pooling mit JDBC 2.0 Extension API [Ste02]

5.3.7.3 Prepared Statements

SQL-Anweisungen, die an eine Datenbank gesendet werden, haben bis zur tatsächlichen Ausführung einige Umwandlungen vor sich. Beginnend mit einer Prüfung auf syntaktische Korrektheit, werden sie danach in einen internen Ausführungsplan der Datenbank übersetzt und mit anderen Transaktionen optimiert. Vorteilhaft wäre es, diese Vorbereitungsarbeiten nur einmal durchführen zu müssen.

So genannte *Prepared Statements* bieten diese Eigenschaft. Sie sind parametrisierte SQL-Anweisungen, die zunächst deklariert und zum Vorkompilieren an eine Datenbank übergeben werden. Später können die Statements beliebig oft ausgeführt werden, indem die formalen Parameter durch aktuelle Werte ersetzt werden und die so parametrisierte Anweisung an die Datenbank übergeben wird. Ein Laufzeitvorteil macht sich besonders dann bemerkbar, wenn zum Beispiel Schleifen Änderungen an Tabellenspalten durchführen.

JDBC stellt Prepared Statements mit dem Interface `PreparedStatement`, das aus `Statement` abgeleitet ist, zur Verfügung. Die Methode `prepareStatement` des Connection-Objekts (Zeile 2) liefert ein `PreparedStatement`.

```
1 Connection con = dataSource.getConnection();
2 PreparedStatement updateMessage = con.prepareStatement(
3 "UPDATE messages SET subject=?, body=? WHERE id=?");
4 updateMessage.setString(1, "subject");
5 updateMessage.setString(2, "body");
6 updateMessage.setInt(3, 5);
7 updateMessage.executeUpdate();
```

Listing 5.8: Beispiel für Prepared Statement

5.4 Fazit

Struts bietet ein solides Grundgerüst für die Entwicklung von Web-Applikationen und eignet sich besonders für Anwendungen mit hohem Anteil an Anwendungslogik.

Die Verwendung des MVC Frameworks ist durch den hohen Gewinn an Übersicht und Wartbarkeit in so gut wie jedem Projekt empfehlenswert. Nur bei sehr kleinen Anwendungen kann es sinnvoll sein, auf MVC zu verzichten, da ein Einsatz von Struts anfangs mit relativ viel Aufwand verbunden ist – viele Komponenten müssen in Java implementiert werden.

Neben Struts haben sich mittlerweile eine Vielzahl weiterer Frameworks etabliert, Beispiele hierfür sind:

- Spring (J2EE)[8]
- Tapestry[9]
- WebWork[10]
- Cocoon[11]
- Turbine[12]
- Appfuse (Meta-Framework)[13]

Literatur: [Fro], [Sch04a], [Kun03], [Wik], [For04], [TNHW03], [str], [Ste02], [Ull04], [Krü04], [Bee03], [Ack04], [dASF01]

[8] http://www.springframework.org
[9] http://jakarta.apache.org/tapestry/
[10] http://www.opensymphony.com/webwork/
[11] http://cocoon.apache.org
[12] http://jakarta.apache.org/turbine/
[13] https://appfuse.dev.java.net

Kapitel 6

Simple SCORM Player

In diesem Kapitel werden zunächst überblicksmäßig einzelne Module der implementierten Web-Applikation „Simple SCORM Player“ vorgestellt. Weiters wird kurz auf das in der Einleitung erörterte Konzept hinsichtlich Tutoring eingegangen. Folgend werden die Architektur und Details zur Implementierung erläutert. Abschließend wird die Integration von SCORM innerhalb der Applikation dargestellt.

An dieser Stelle sei erwähnt, dass der Implementierung der Lernumgebung zwei Magisterarbeiten zu Grunde liegen und daher an mehreren Stellen bewusst auf Details verzichtet und auf die entsprechende Arbeit referenziert wird.

6.1 Requirements

In diesem Abschnitt werden funktionale Anforderungen an das zu implementierende Learning-Management-System erläutert. Ausgehend von den gewonnenen Erkenntnissen in Kapitel 4 auf Seite 78 erweisen sich folgende Anforderungen und Basis-Funktionalitäten als sinnvoll:

Architektur/Aufbau: Das zu implementierende System soll einen modularen Aufbau besitzen – einerseits um eine Arbeitsaufteilung innerhalb eines Entwicklerteams zu ermöglichen, andererseits um etwaige (nachträgliche) Modifikationen effizient durchführen zu können. Weiters wird die Verwendung eines entsprechenden Frameworks und Open Source-Software gefordert. Die

zu implementierende Applikation soll Web-basiert sein und gängigen W3C-Standards entsprechen.

Benutzerverwaltung: Die Verwaltung von Benutzern soll Rollen-basiert sein und dabei zumindest zwischen zwei Benutzergruppen wie Studenten und Tutoren unterscheiden können. Weiters soll die Anwendung übliche Verwaltungsfunktionen wie Anlegen, Editieren und Löschen von Benutzern umfassen.

Rechtemanagement: Abhängig von den Benutzerrollen sollen Benutzer individuelle Rechte innerhalb des Systems aufweisen. Eine unterschiedliche Rechtevergabe ist überlicherweise in der Verwaltung von Benutzern und Kursen sowie in einem Diskussionsforum sinnvoll.

Kursverwaltung: Kurse sollten übersichtlich angezeigt und von Studenten ausgeführt werden können. Eine Importmöglichkeit von Standard-konformen Kurseinheiten seitens des Tutors soll ebenfalls möglich sein. Weiters sollen bestehende Kurse modifizierbar sein, indem Kursinhalte hinzugefügt oder entfernt werden können.

Kommunikationswerkzeuge: Zumindest eine asynchrone Möglichkeit der Kommunikation wie zum Beispiel ein Diskussionsforum muss unterstützt werden. Hierbei soll das Forum typische Eigenschaften wie Erstellen, Editieren, Löschen und Bestimmen der Sichtbarkeit von Nachrichten aufweisen.

Authoring: Mittels integrierten Authoring-Tools soll einem Tutor die Erstellung einfacher Kursinhalte ermöglicht werden; eine Unterstützung gängiger Dateiformate wird vorausgesetzt.

Unterstützung von Standards: Das LMS muss eine rudimentäre Unterstützung der aktuellsten SCORM-Spezifikation beinhalten und neben einem Kursimport zumindest einfache Sequencing-Regeln interpretieren können.

Individuelles Tutoring/Adaptierung: Mittels einer Anpassung von Lernpfaden und einfachen Authoring-Möglichkeiten soll eine effiziente Unterstützung für Studenten während des Ausführens von Kursen gegeben sein. Weiters soll die Möglichkeit geboten werden, diesen Prozess zu automatisieren.

Internationalisierung: Eine Unterstützung von zumindest zwei Sprachen (deutsch, englisch) soll implementiert sein; ein Hinzufügen weiterer Sprachmodule soll ebenfalls möglich sein.

6.2 Module

Der Simple SCORM Player, in weiterer Folge auch mit SSP abgekürzt, ist eine Web-Applikation mit dem Hauptziel, die Basisfunktionalität eines Learning-Management-Systems zu implementieren. Im folgenden werden zunächst elementare Module hinsichtlich ihrer Funktionen beschrieben, ohne näher auf Implementierungsdetails einzugehen.

Die Web-Applikation ist Rollen-basiert und unterstützt dabei die Rollen des Studenten und die eines Tutors. Abhängig von den Benutzerrollen sind auch Funktionalitäten einzelner Module innerhalb des SSP angepasst; dementsprechend wird bei der Modul-Beschreibung darauf Rücksicht genommen.

6.2.1 Anmeldung

Dieses Modul (siehe Abb. 6.1 auf der folgenden Seite) beinhaltet einen typischen Login-Vorgang in das Learning-Management-System, erweitert um zusätzlichen Funktionen. Die Anmeldung erfolgt für Studenten und Tutoren auf die gleiche Weise:

- *Anmelden eines Benutzers* - Mittels Eingabe des Benutzernamens und des Passwortes erfolgt ein Login-Vorgang. Bei Fehlschlagen aufgrund Eingabe falscher Daten wird eine Fehlermeldung auf der Startseite ausgegeben und der Vorgang muss wiederholt werden.
- *Registrieren eines neuen Benutzers* - An dieser Stelle ist es möglich, sich als neuer Kursteilnehmer (Student) zu registrieren. Nach erfolgreichem Abschließen des Registrierungsvorganges erfolgt ein automatischer Login in das System. Ein neu erstelltes Benutzerprofil umfasst folgende Attribute: Benutzername, Vorname, Nachname, E-Mail, Geschlecht, Geburtsdatum und Passwort.

  Um einen Benutzer im System eindeutig zuordnen zu können, ist die Angabe des Benutzernamens eindeutig; falls bereits ein entsprechender User registriert ist, wird es dem Benutzer mitgeteilt.

  Die Erstellung eines neuen Tutors ist nur innerhalb des SSP möglich und wird in Abschnitt 6.2.2 auf der nächsten Seite erläutert.

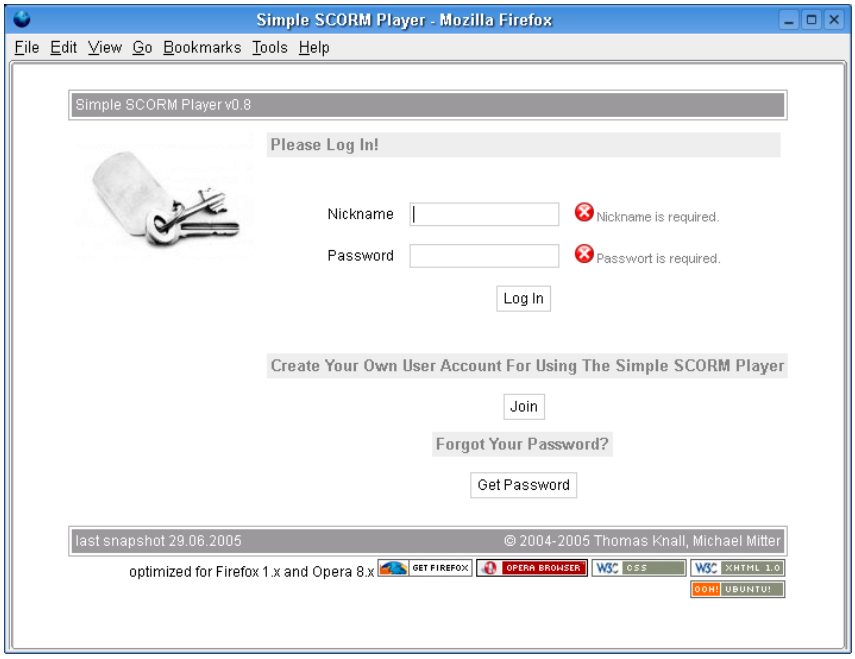


Abb. 6.1: Login-Ansicht

- *Wiederfinden eines vergessenen Passwortes* - Sollten Account-Daten vergessen worden sein, besteht hier die Möglichkeit, mittels gültiger E-Mail-Adresse die Zugangsdaten seines Accounts via Mail zu erhalten.

6.2.2 Benutzerverwaltung

Dieses Modul umfasst eine komplette Rollen-basierte Benutzerverwaltung (siehe Abb. 6.2 auf der folgenden Seite) mit folgenden Möglichkeiten:

Tutor: Bei Aufrufen der Benutzerverwaltung werden dem Tutor sämtliche im System registierte Benutzer aufgelistet. Neben den bereits erwähnten Benutzerattributen werden die jeweilige Benutzerrolle und zusätzlich Statistiken wie Datum der Registrierung, Anzahl der Logins sowie Zeit und Datum des letzten Logins angezeigt. Das Modul beinhaltet folgende Funktionen:

- *Anlegen neuer Benutzer* - Tutoren können neue Benutzerprofile erstellen (siehe Abb. 6.3 auf Seite 140). Im Gegensatz zum Registrieren eines Benutzers beim Login-Vorgang besteht hier auch die Möglichkeit, Benutzerrollen individuell festzulegen.

- *Editieren vorhandener Benutzer* - Attribute von registrierten Benutzern können hier verändert werden. Der Benutzername ist jedoch eindeutig und kann daher keinen Änderungen unterzogen werden.
- *Löschen von Benutzer* - Tutoren können eine Löschung registrierter Benutzer vornehmen, der aktuell eingeloggte Tutor ist davon ausgenommen. Weiters werden beim Löschvorgang mit dem Benutzer verbundene Nachrichten innerhalb des Diskussionsforums (siehe Abschnitt 6.2.5 auf Seite 145) und benutzerbezogene Lerneinheiten (siehe Abschnitt 6.3 auf Seite 149) dahingehend verändert, dass diese nicht mehr angezeigt werden oder mittels entsprechendem Hinweis auf die Löschung des Benutzers hingewiesen wird.

Student:

- *Editieren des Benutzerprofils* - Ein angemeldeter Student hat die Möglichkeit, sein Benutzerprofil – den Benutzernamen ausgenommen – beliebig zu ändern.

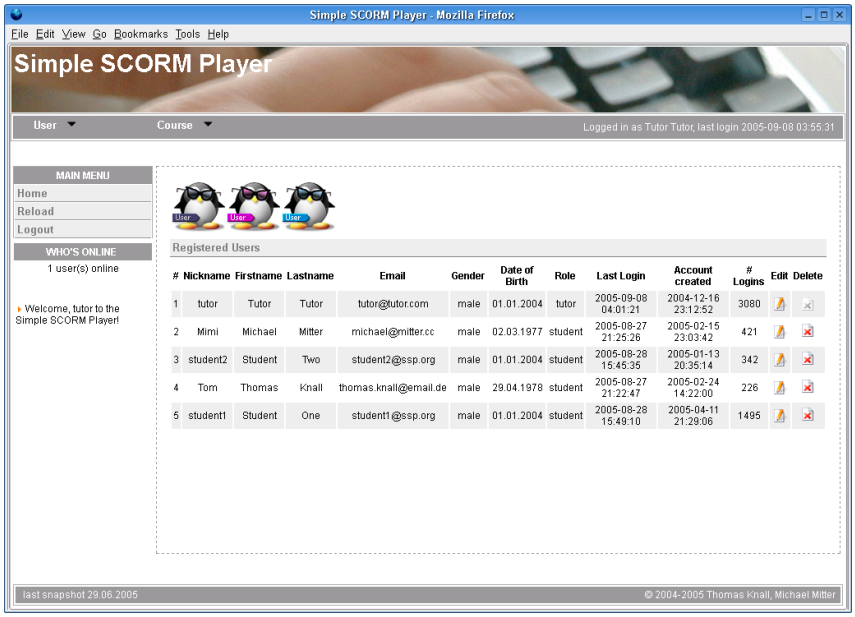

Simple SCORM Player - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Simple SCORM Player
User
Course
Logged in as Tutor Tutor, last login 2005-09-08 03:55:31
MAIN MENU
Home
Reload
Logout
WHO'S ONLINE
1 user(s) online
Welcome, tutor to the Simple SCORM Player!
Registered Users
Nickname Firstname Lastname Email Gender Date of Birth Role Last Login Account created # Logins Edit Delete
1 tutor Tutor Tutor tutor@tutor.com male 01.01.2004 tutor 2005-09-08 04:01:21 2004-12-16 23:12:52 3080
2 Mimi Michael Mitter michael@mitter.cc male 02.03.1977 student 2005-08-27 21:25:26 2005-02-15 23:03:42 421
3 student2 Student Two student2@ssp.org male 01.01.2004 student 2005-08-28 15:45:35 2005-01-13 20:35:14 342
4 Tom Thomas Knall thomas.knall@email.de male 29.04.1978 student 2005-08-27 21:22:47 2005-02-24 14:22:00 226
5 student1 Student One student1@ssp.org male 01.01.2004 student 2005-08-28 15:49:10 2005-04-11 21:29:06 1495
last snapshot 29.06.2005
© 2004-2005 Thomas Knall, Michael Mitter

Abb. 6.2: Benutzerverwaltung

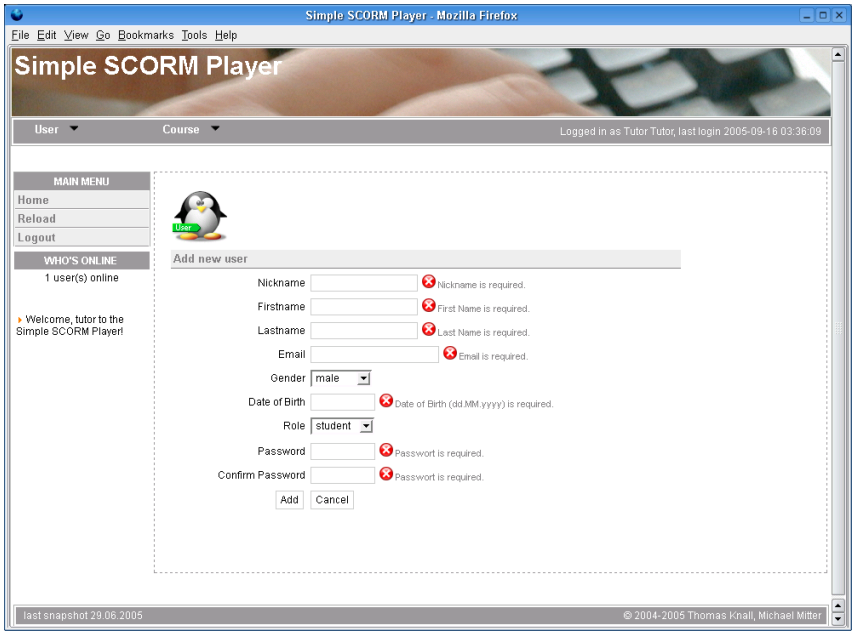


Abb. 6.3: Maske für das Erstellen eines neuen Benutzers

6.2.3 Kursverwaltung

Das Modul Kursverwaltung (siehe Abb. 6.5 auf Seite 142) umfasst mehrere Sub-Module. Bei Aufruf des Kurs-Moduls erfolgt zunächst eine Auflistung aller dem System zur Verfügung stehenden Kurse. Neben Kurstitel werden auch das Importdatum und Informationen, das Sequencing (siehe Abschnitt 6.4.4.2 auf Seite 161) betreffend, aufgelistet. Weiters wird die Anzahl kurszugehöriger Nachrichten angezeigt. Folgende Aktionen sind möglich:

Tutor:

- *Starten eines Kurses* - Nach Starten eines Kurses werden das Kursmenü, entsprechend den Sequencing-Attributen, und eine Startseite mit dem Kurstitel geladen.
- *Kurs Upload* - Unter diesem Menüpunkt können SCORM-konforme Kurspakete in das System geladen werden. Während des Importvorgangs wird das Paket entpackt und in ein Verzeichnis des Simple SCORM Player kopiert. An dieser Stelle erfolgt mittels modifiziertem ADL-Parser auch eine entsprechende Validierung des SCORM-Kurses (siehe Abschnitt 6.4.4.1 auf Seite 160); bei Nichtkonformität oder falschem Da-

teiformat werden entsprechende Fehlermeldungen ausgegeben und der Importvorgang muss wiederholt werden.

- *Kurslektionen* - Der Simple SCORM Player umfasst mehrere Möglichkeiten, Kurslektionen hinzuzufügen, zu löschen und zu editieren (siehe Abschnitt 6.2.4 auf Seite 143).
- *Sequencing* - Innerhalb des SSP besteht die Möglichkeit, Sequencing-Attribute (siehe Abschnitt 6.4.4.2 auf Seite 161) festzulegen und so die Navigation innerhalb eines Kurses zu bestimmen.

Student:

- *Starten eines Kurses* - Nach Starten eines Kurses wird das Kursmenü entsprechend den Sequencing-Attributen geladen, alle vom Tutor freigeschaltenen Kurslektionen werden angezeigt (siehe Abb. 6.4). Der Student kann mit dem Ausführen des Kurses beginnen.

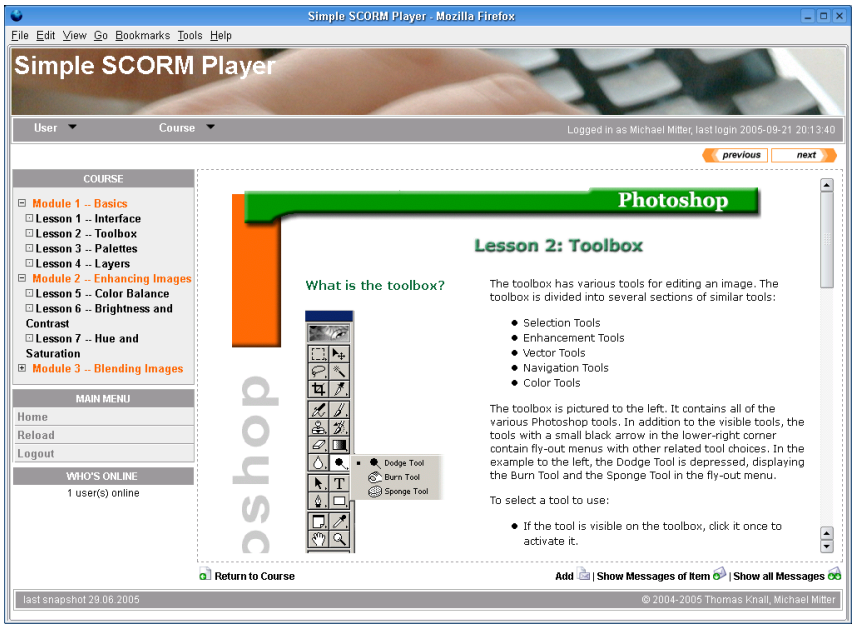

Simple SCORM Player - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Simple SCORM Player
User
Course
Logged in as Michael Mitter, last login 2005-09-21 20:13:40
previous
next
COURSE
Module 1 -- Basics
Lesson 1 -- Interface
Lesson 2 -- Toolbox
Lesson 3 -- Palettes
Lesson 4 -- Layers
Module 2 -- Enhancing Images
Lesson 5 -- Color Balance
Lesson 6 -- Brightness and Contrast
Lesson 7 -- Hue and Saturation
Module 3 -- Blending Images
MAIN MENU
Home
Reload
Logout
WHO'S ONLINE
1 user(s) online
Photoshop
Lesson 2: Toolbox
What is the toolbox?
The toolbox has various tools for editing an image. The toolbox is divided into several sections of similar tools:
Selection Tools
Enhancement Tools
Vector Tools
Navigation Tools
Color Tools
The toolbox is pictured to the left. It contains all of the various Photoshop tools. In addition to the visible tools, the tools with a small black arrow in the lower-right corner contain fly-out menus with other related tool choices. In the example to the left, the Dodge Tool is depressed, displaying the Burn Tool and the Sponge Tool in the fly-out menu.
To select a tool to use:
If the tool is visible on the toolbox, click it once to activate it.
Dodge Tool
Burn Tool
Sponge Tool
Return to Course
Add | Show Messages of Item | Show all Messages
last snapshot 29.06.2005
© 2004-2005 Thomas Knaill, Michael Mitter

Abb. 6.4: Ausführen eines Kurses

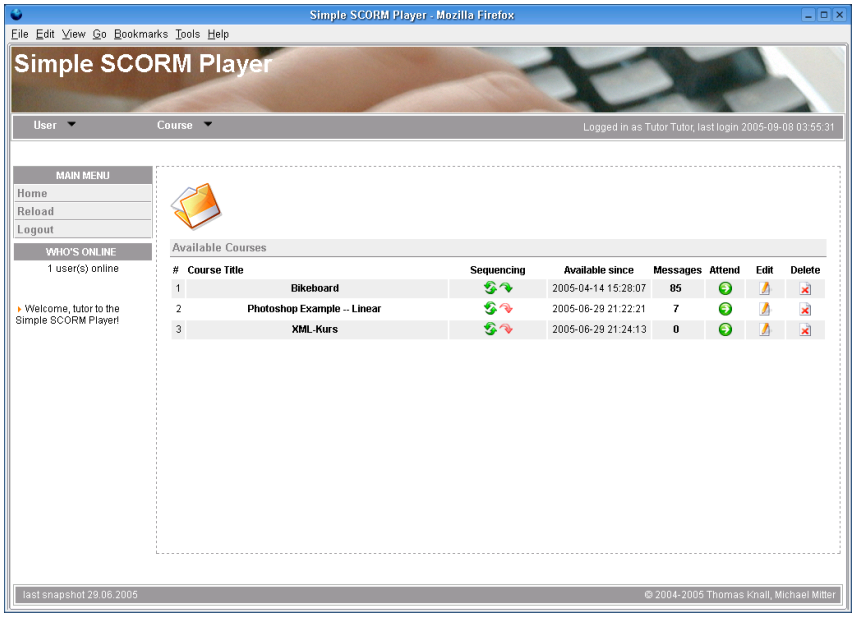

Simple SCORM Player - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Simple SCORM Player
User
Course
Logged in as Tutor Tutor, last login 2005-09-08 03:55:31
MAIN MENU
Home
Reload
Logout
WHO'S ONLINE
1 user(s) online
Welcome, tutor to the Simple SCORM Player!
Available Courses
Course Title Sequencing Available since Messages Attend Edit Delete
1 Bikeboard 2005-04-14 15:28:07 85
2 Photoshop Example -- Linear 2005-06-29 21:22:21 7
3 XML-Kurs 2005-06-29 21:24:13 0
last snapshot 29.06.2005
© 2004-2005 Thomas Knaill, Michael Mitter

Abb. 6.5: Kursübersicht

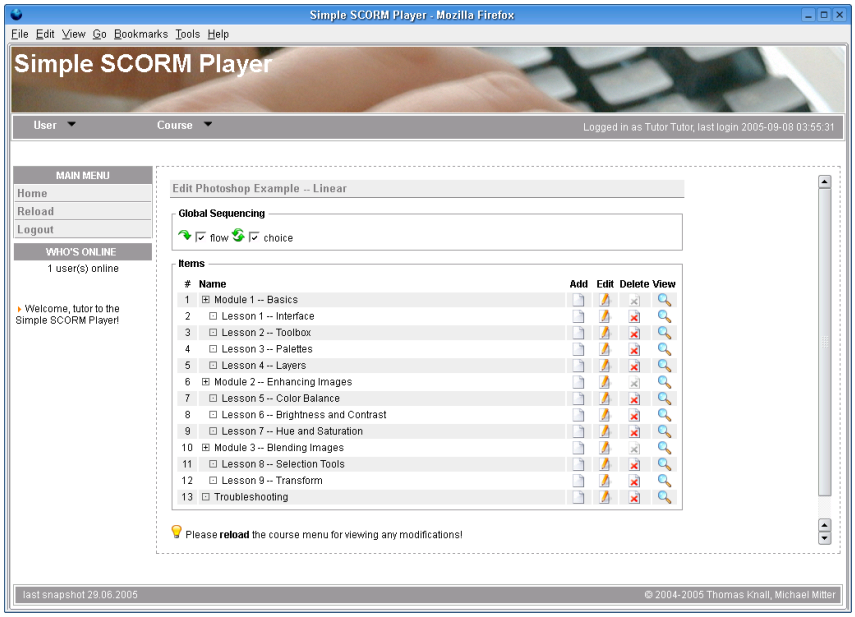

Simple SCORM Player - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Simple SCORM Player
User
Course
Logged in as Tutor Tutor, last login 2005-09-08 03:55:31
MAIN MENU
Home
Reload
Logout
WHO'S ONLINE
1 user(s) online
Welcome, tutor to the Simple SCORM Player!
Edit Photoshop Example -- Linear
Global Sequencing
flow choice
Items
Name Add Edit Delete View
1 Module 1 -- Basics
2 Lesson 1 -- Interface
3 Lesson 2 -- Toolbox
4 Lesson 3 -- Palettes
5 Lesson 4 -- Layers
6 Module 2 -- Enhancing Images
7 Lesson 5 -- Color Balance
8 Lesson 6 -- Brightness and Contrast
9 Lesson 7 -- Hue and Saturation
10 Module 3 -- Blending Images
11 Lesson 8 -- Selection Tools
12 Lesson 9 -- Transform
13 Troubleshooting
Please reload the course menu for viewing any modifications!
last snapshot 29.06.2005
© 2004-2005 Thomas Knaill, Michael Mitter

Abb. 6.6: Editieren eines Kurses

6.2.4 Kurslektionen (Items)

In der Editieransicht einer Kurseinheit (siehe Abb. 6.8 auf Seite 145) werden zunächst alle Elemente [1] eines Kurses – vielfach spricht man auch von Items oder Lektionen – ihrer logischen Struktur folgend aufgelistet. Dabei wird zwischen Container-Elementen und einfachen Kurselementen unterschieden. Kurselementen können typischen SCORM-Lektionen oder einfachen Textnachrichten entsprechen. Es stehen folgende Funktionen zur Verfügung:

Tutor:

- *Editieren* - Je nach Typ der jeweiligen Kurslektion unterscheiden sich auch die zu editierenden Attribute. Neben Name können auch referenzierte Dateien oder Nachrichten modifiziert werden. Für Dateien ist ein üblicher Upload-Dialog integriert.

  Weiters besteht die Möglichkeit, die Sichtbarkeit des jeweiligen Kurselements individuell für jeden Studenten festzulegen. Dabei stehen zwei Auswahllisten zur Verfügung, mit welchen Studenten entsprechend der gewünschten Sichtbarkeit zugeordnet werden können. Bei Festlegen der Sichtbarkeit ist auch zu berücksichtigen, ob es sich um ein einzelnes Kurselement oder einen Container handelt.

  In diesem Dialog existiert auch die Möglichkeit, so genannte Tutoraktionen – in Zusammenhang mit dem Simple SCORM Player auch „Activities“ genannt – festzulegen. Dieser Vorgang wird in Abschnitt 6.3 auf Seite 149 genauer beschrieben.

- *Hinzufügen* - Bei Hinzufügen einer Lektion besteht die Möglichkeit, den Typ einer Lektion zu bestimmen. Weiters ist die Position des hinzuzufügenden Kurselements anzugeben. Zwischen folgenden Typen wird innerhalb des SSP unterschieden:
  - *Container* - Entsprechend der SCORM-Spezifikation besteht die Möglichkeit, ein neues Container-Element anzulegen. Dabei sind Name und Position anzugeben. Container verfügen über keine weiteren Elemente.
  - *Item (Lektion)* - Neben einem Namen ist noch ein zugehöriges File anzugeben. Dieses wird mittels einem Upload-Dialog ins entspre-

[1] Eine Definition möglicher Kurselemente erfolgt in Abschnitt 6.3.2 auf Seite 150.

chende Kursverzeichnis innerhalb des Systems geladen und kann typischen Dokumentformaten entsprechen.

– *Frage/Kommentar* - Lektionen können auch einfachen Textnachrichten entsprechen. Der Typ, Frage oder Kommentar, wird nach Anlegen in der Menüstruktur des Kurses entsprechend gekennzeichnet und kann so von standardmäßigen Kurslektionen unterschieden werden.

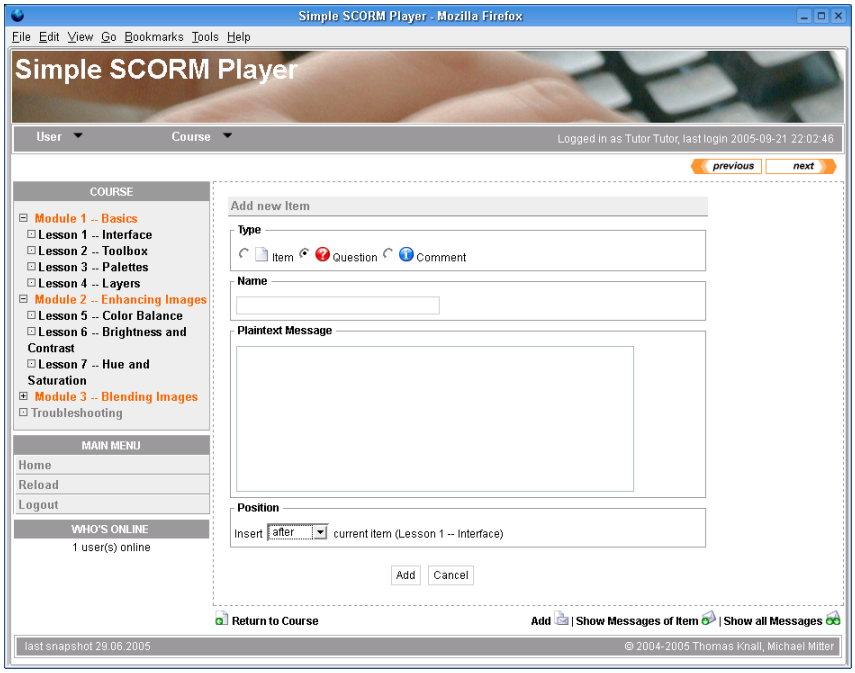


Abb. 6.7: Hinzufügen einer Kurslektion

- *Löschen* - Es kann eine einzelne Lektion oder ein ganzer Container gelöscht werden. Für letzteren bedeutet dies, dass auch alle Unterelemente mitgelöscht werden. Weiters werden entsprechend zugehörige Nachrichten im Diskussionsforum ebenfalls mitausgeblendet.
- *Vorschau* - Aus Usability-Gründen besteht für einen Tutor die Möglichkeit, den jeweiligen Inhalt einer Kurseinheit via eines Popup-Fenster anzuzeigen. Somit erspart man sich ein unnötiges Herumnavigieren zwischen der verschiedenen Ansichten.

Student: Studenten sind von diesen Editier-Funktionalitäten ausgeschlossen.

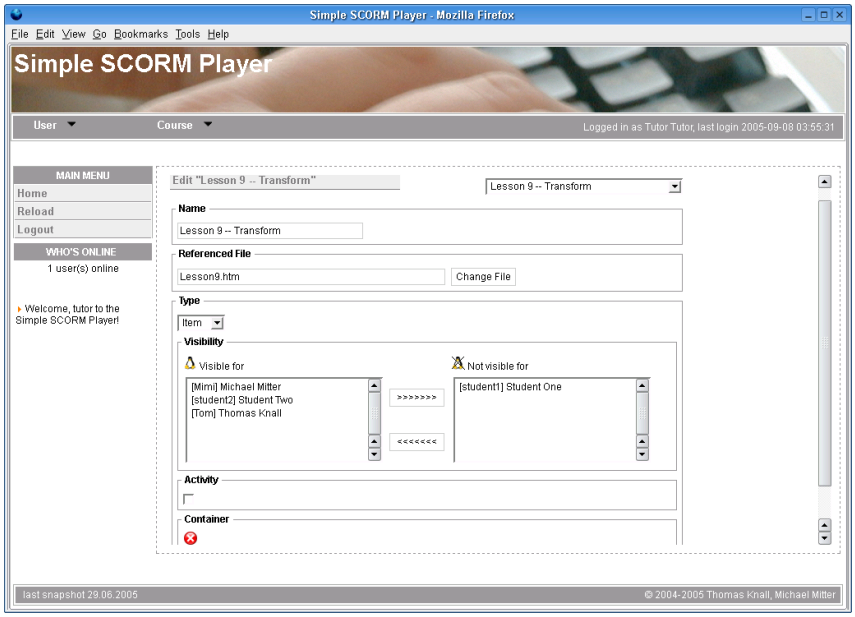


Abb. 6.8: Editieren einer Kurslektion

6.2.5 Diskussionsforum

Das Forum oder im Simple SCORM Player auch *MessageBoard* (siehe Abb. 6.10 auf Seite 148) genannt bietet Studenten eine Möglichkeit, mit Tutoren aber auch untereinander Informationen zu einzelnen Lektionen auszutauschen. Bei Aufruf des MessageBoards werden in der Gesamtansicht, Threads nach ihrem Erstellungsdatum sortiert, angezeigt. Weiters werden Betreff, zugehörige Lektion, Nachrichtentyp, Anzahl der Antworten, Erstellung und Datum der zuletzt hinzugefügten Nachricht aufgelistet. Bei der Ansicht kann zwischen Lektions- und Kurs-zugehörigen Nachrichten unterschieden werden. Das Modul beinhaltet folgenden Funktionsumfang:

Tutor:

- *Neue Nachricht* - Das Hinzufügen einer neuen Nachricht wird typischerweise mittels einem Formular ermöglicht. Hier können neben Betreff und Nachrichtentext noch der Typ (Frage oder Kommentar) sowie die Sichtbarkeit bestimmt werden.
- *Antworten* - Bei Verfassen einer Antwort wird einerseits automatisch die Sichtbarkeit der Originalnachricht berücksichtigt, sowie der Typ der Ant-

wort vorgeschlagen.

- *Editieren* - Hier wird ebenfalls die Sichtbarkeit berücksichtigt und die editierte Nachricht mit einem Vermerk des Autors und Datum versehen.
- *Löschen* - Ein Löschen von Nachrichten ist nur dann möglich, wenn ein Thread nur mehr aus einer einzigen Nachricht besteht. Somit wird verhindert, dass verwaiste Nachrichten im Diskussionsforum aufscheinen.
- *Finden einer ähnlichen Nachricht* - Hier besteht vor allem für Studierende die Möglichkeit, automatisch ähnliche Nachrichten innerhalb des Diskussionsforums zu suchen. Bei einem Erfolg, werden gefundene Nachrichten ähnlich wie im MessageBoard aufgelistet. Zusätzlich wird dem Benutzer noch der Ähnlichkeitsgrad der entsprechenden Nachricht angezeigt.
- *Erstellen einer Aktion* - Der Tutor wird dabei an den Editier-Dialog weitergeleitet, um dort entsprechende Tutoraktionen (Abb. 6.9) zur jeweiligen Nachricht zu setzen.

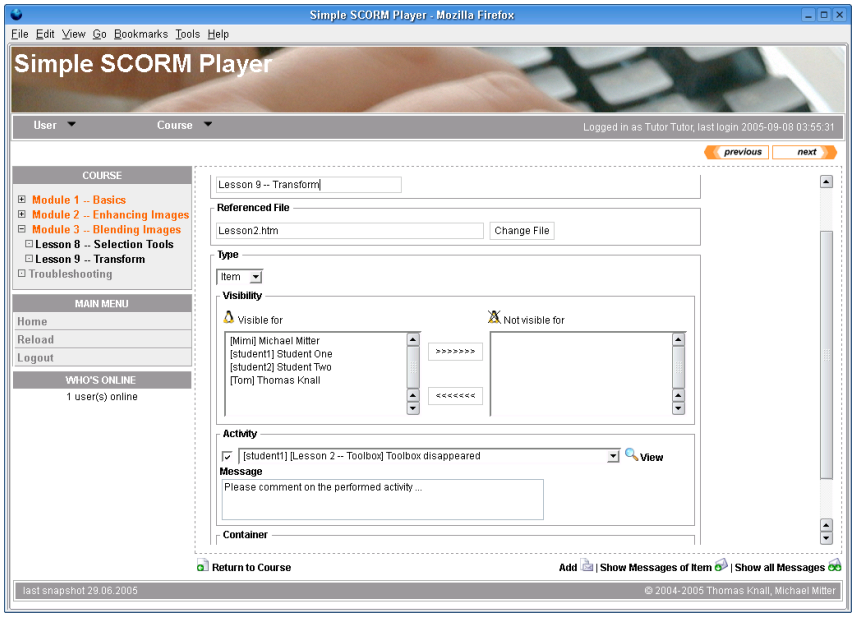


Abb. 6.9: Erstellen einer Aktion

Student: Prinzipiell entsprechen die Funktionen jenen des Tutors jedoch mit einigen Einschränkungen.

- *Neue Nachricht* - Das Hinzufügen einer neuen Nachricht (siehe Abb. 6.11 auf der nächsten Seite) wird typischerweise mittels einem Formular ermöglicht. Hier können neben Betreff und Nachrichtentext noch der Typ (Frage oder Kommentar) sowie die Sichtbarkeit bestimmt werden. Die Möglichkeit, eine Nachricht zu verfassen, hängt von der Sichtbarkeit einzelner Lektionen oder Nachrichten im MessageBoard ab.
- *Antworten* - Bei Verfassen einer Antwort wird einerseits automatisch die Sichtbarkeit der Original-Nachricht berücksichtigt, sowie der Typ der Antwort vorgeschlagen. Durch automatisches Festlegen der Sichtbarkeit wird verhindert, das beispielsweise Antworten auf nicht sichtbare Fragen existieren können.
- *Editieren* - Hier wird ebenfalls die Sichtbarkeit berücksichtigt und die geänderte Nachricht mit einem Vermerk des Autors und dem Datum versehen. Studenten können nur selbst verfasste Nachrichten editieren.
- *Löschen* - Ein Löschen von Nachrichten ist nur dann möglich, wenn ein Thread nur mehr aus einer einzigen Nachricht besteht. Studenten können wiederum nur selbst verfasste Nachrichten löschen.
- *Finden einer ähnlichen Nachricht* - Hier besteht die Möglichkeit, automatisch ähnliche Nachrichten innerhalb des Diskussionsforums zu suchen. Bei einem Erfolg, werden gefundene Nachrichten ähnlich wie im MessageBoard aufgelistet. Zusätzlich wird dem Benutzer noch der Ähnlichkeitsgrad der entsprechenden Nachricht angezeigt.

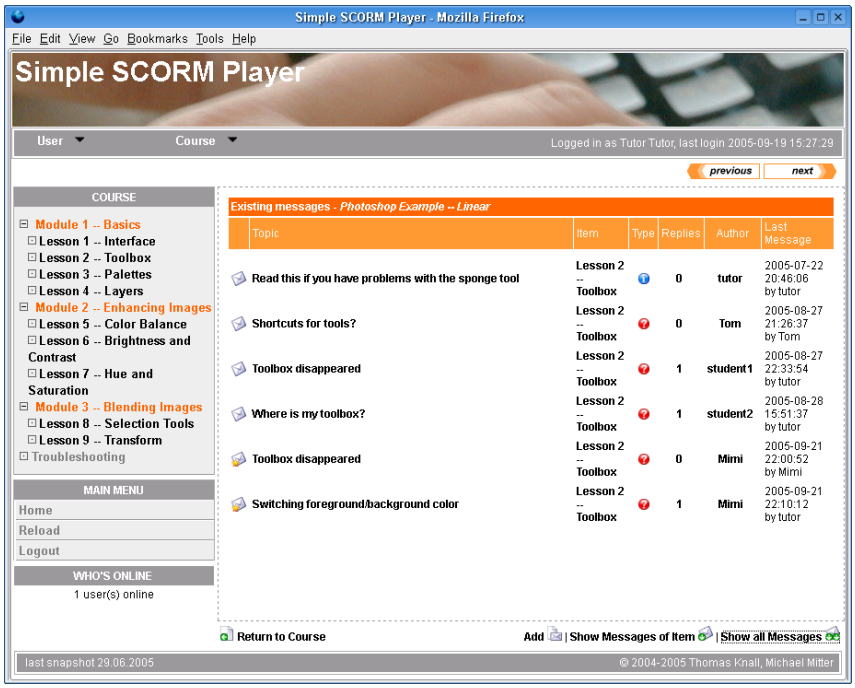


Abb. 6.10: Diskussionsforum (MessageBoard)

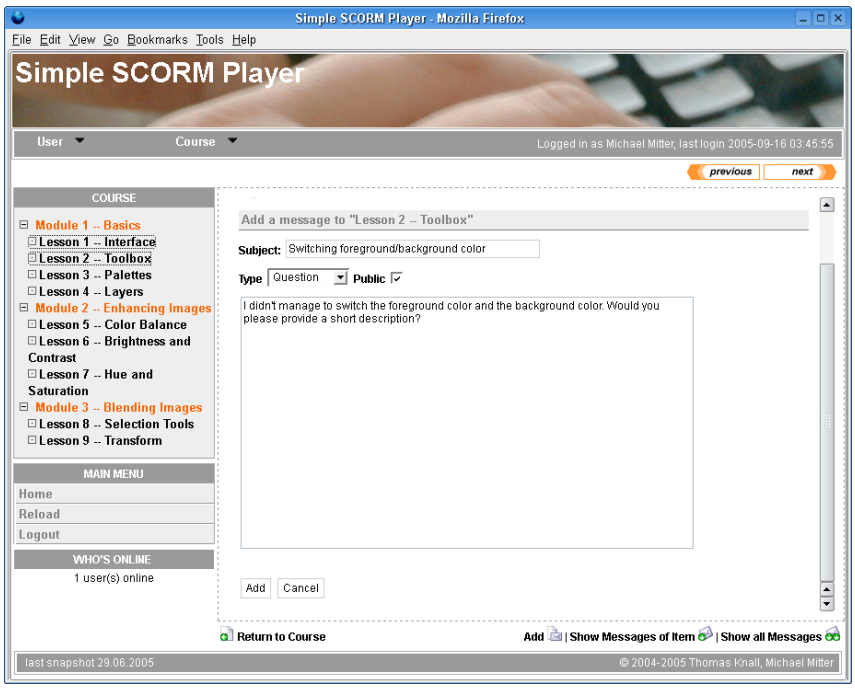

Simple SCORM Player - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Simple SCORM Player
User
Course
Logged in as Michael Mitter, last login 2005-09-16 03:45:55
previous
next
COURSE
Module 1 -- Basics
Lesson 1 -- Interface
Lesson 2 -- Toolbox
Lesson 3 -- Palettes
Lesson 4 -- Layers
Module 2 -- Enhancing Images
Lesson 5 -- Color Balance
Lesson 6 -- Brightness and Contrast
Lesson 7 -- Hue and Saturation
Module 3 -- Blending Images
Lesson 8 -- Selection Tools
Lesson 9 -- Transform
MAIN MENU
Home
Reload
Logout
WHO'S ONLINE
1 user(s) online
Add a message to "Lesson 2 -- Toolbox"
Subject: Switching foreground/background color
Type Question Public
I didn't manage to switch the foreground color and the background color. Would you please provide a short description?
Add
Cancel
Return to Course
Add | Show Messages of Item | Show all Messages
last snapshot 29.06.2005
© 2004-2005 Thomas Knall, Michael Mitter

Abb. 6.11: Hinzufügen einer neuen Nachricht

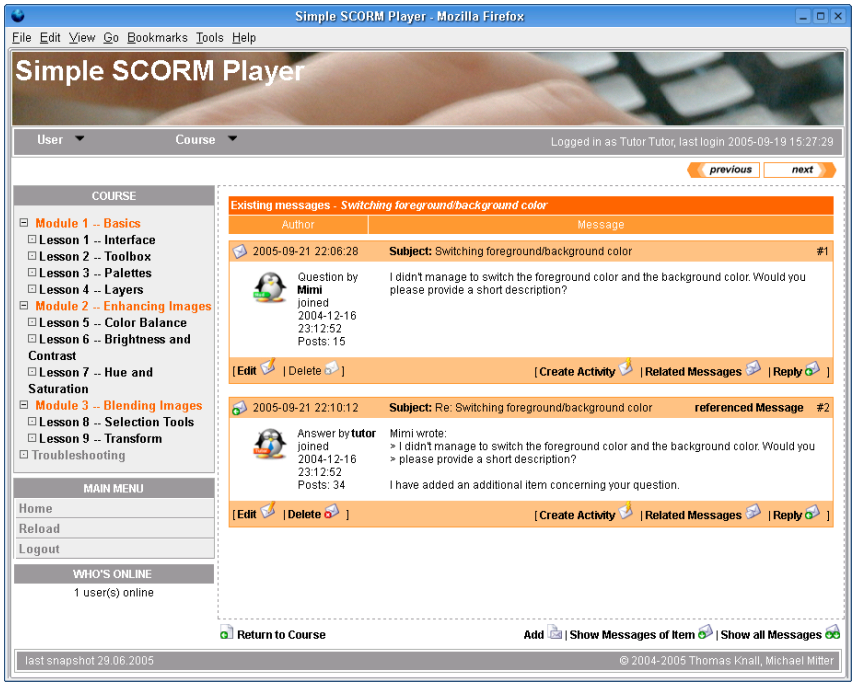

Simple SCORM Player - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Simple SCORM Player
User
Course
Logged in as Tutor Tutor, last login 2005-09-19 15:27:29
previous
next
COURSE
Module 1 -- Basics
Lesson 1 -- Interface
Lesson 2 -- Toolbox
Lesson 3 -- Palettes
Lesson 4 -- Layers
Module 2 -- Enhancing Images
Lesson 5 -- Color Balance
Lesson 6 -- Brightness and Contrast
Lesson 7 -- Hue and Saturation
Module 3 -- Blending Images
Lesson 8 -- Selection Tools
Lesson 9 -- Transform
Troubleshooting
MAIN MENU
Home
Reload
Logout
WHO'S ONLINE
1 user(s) online
Existing messages - Switching foreground/background color
Author
Message
2005-09-21 22:06:28
Subject: Switching foreground/background color
#1
Question by Mimi joined 2004-12-16 23:12:52 Posts: 15
I didn't manage to switch the foreground color and the background color. Would you please provide a short description?
[Edit | Delete]
[Create Activity | Related Messages | Reply]
2005-09-21 22:10:12
Subject: Re: Switching foreground/background color
referenced Message
#2
Answer by tutor joined 2004-12-16 23:12:52 Posts: 34
Mimi wrote:
> I didn't manage to switch the foreground color and the background color. Would you
> please provide a short description?
I have added an additional item concerning your question.
[Edit | Delete]
[Create Activity | Related Messages | Reply]
Return to Course
Add | Show Messages of Item | Show all Messages
last snapshot 29.06.2005
© 2004-2005 Thomas Knall, Michael Mitter

Abb. 6.12: MessageBoard, Thread-Ansicht

6.3 Individuelles Tutoring

Ziel bei der Entwicklung des Simple SCORM Player war es auch, Konzepte für ein individuelles Tutoring von Studenten zu finden. Die grundlegende Idee besteht darin, durch einfache Änderungen innerhalb der Applikation, Lernpfade anpassen zu können. Im Kontext mit E-Learning spricht man auch von „persönlichem Tutoring“ (siehe Abschnitt 2.4.3 auf Seite 20).

An dieser Stelle soll natürlich nicht unerwähnt bleiben, dass innerhalb dieser Arbeit aufgrund der Komplexität nur jeweils Ansätze diskutiert und implementiert werden können.

6.3.1 Problemstellung

Bei der Evaluierung von Learning-Management-Systemen in Kapitel 4 auf Seite 78 boten viele Systeme einen hohen Grad an Funktionalität, dennoch war es oftmals nicht möglich, in nur kurzer Zeit eine Kurslektion anzupassen oder hinzuzufügen.

Eine Unterstützung hinsichtlich Standards – hier vor allem von SCORM – war großteils auch vorhanden, nur nach Import einer fertigen Kurseinheit, war ein nachträgliches Anpassen teilweise nur schwer möglich. Die integrierten Authoring-Tools konzentrierten sich hier vermehrt auf eine Erstellung eigener Kurse.
Die evaluierten Systeme umfassten eine Vielzahl an Kommunikationsmöglichkeiten, sowohl asynchron als auch synchron. Auffallend war, dass sich die implementierten Diskussionsforen nicht explizit an Kurslektionen binden ließen, sondern meistens unabhängig agierten.

6.3.2 Konzept

Bereits in der Einleitung wurde erörtert, welche Möglichkeiten hinsichtlich Effizienz und einfacher Durchführbarkeit hinsichtlich Tutoring für sinnvoll erscheinen. Eine Kombination aus einem Authoring-Werkzeug und einem Diskussionsforum soll eine Anpassung von Lernpfaden ermöglichen.
Für die Erläuterung des Konzeptes sind folgende Begriffe und Definitionen von Bedeutung:

Kurslektion: Eine Lektion – innerhalb von SCORM ein so genanntes *Item* – entspricht einer typischen Lerneinheit in einem Kurs. Der SCORM-Spezifikation zur Folge ist diese mit einer externen Ressource verbunden. Prinzipiell kann es sich dabei um jedes beliebige Dateiformat handeln. Im Simple SCORM Player hat es sich als sinnvoll erwiesen, aufgrund der Darstellung innerhalb des Browsers HTML-Dokumente, Abbildungen und Plaintext-Dateien zu verwenden. Somit entgeht man einer eventuellen Abhängigkeit von Plugins oder extern auszuführenden Programmen.

Container stellen übergeordnete Elemente innerhalb einer Kursstruktur dar. Ihnen ist eine beliebige Anzahl an Kurslektionen untergeordnet, sie können aber auch selbst eine beliebige Anzahl an Containern enthalten. Entsprechend der SCORM-Spezifikation (siehe [ADL04b]) dürfen Container nicht mit externen Ressourcen verknüpft werden.

Kurs: Ein SCORM-Kurs entspricht einer beliebigen Anzahl an Containern und Lektionen. In Abb. 3.8 auf Seite 44 wird die prinzipielle Struktur dargestellt.

Lernpfade entsprechen in SCORM der so genannten Content Hierachy des Content Packaging. Dabei sind vorgegebene Kurslektionen logisch innerhalb

einer Baumstruktur angeordnet. Die Abfolge der Lernschritte lässt sich mittels Hinzufügen oder Ausblenden einzelner Lektionen steuern.

Aktion: Tutoren besitzen innerhalb des Simple SCORM Player die Möglichkeit, sogenannte Aktionen oder auch „Activities“ zu erstellen und auszuführen. Darunter versteht man ein Ein- und Ausblenden einzelner Lektionen. Aktionen sind immer Nachrichten von Studenten innerhalb des Forums zugeordnet. Für ein späteres Identifizieren wird die jeweilige Aktion mit einem Kommentar versehen.

Authoring: Innerhalb der Applikation besteht für einen Tutor die Möglichkeit, mittels eines einfachen Authoring-Tools, Kurslektionen zu erstellen. Es können damit sowohl fertige Dateien hinzugefügt werden als auch einfache Plaintext-Seiten. Letztere ermöglichen es auch, gezielt Fragen am Ende eines Kurses an Studierende zu richten.

Diskussionforum: Das MessageBoard stellt innerhalb des SSP ein zentrales Element dar. Sämtliche Kommunikation zwischen Studenten und Tutor(-en) erfolgt über dieses Modul. Es ist möglich, Fragen zu einzelnen Lektionen zu stellen oder einfach nur Problemstellungen zu diskutieren. Jede erstellte Nachricht ist immer direkt oder indirekt – z.B. bei Antworten – einer speziellen Lektion innerhalb eines Kurses zugeordnet. Somit lässt sich stets einfach feststellen, welche Threads zu welchen Lektionen zuzuordnen sind.

6.3.3 Möglicher Workflow

Es folgt nun eine kurze Beschreibung eines typisches Workflows bei der Anpassung von Lernpfaden verbunden mit Tutoraktionen. Prinzipiell wird hier ein Interagieren der in Kapitel 6.2 auf Seite 137 vorgestellten Module erläutert.

Auf weitere Screenshots wird aus Gründen der Redundanz und Unübersichtlichkeit verzichtet. Es wird an den entsprechenden Stellen auf die jeweiligen Abbildungen in den vorangegangenen Kapiteln verwiesen.

1. Ein Student wählt in der Kursverwaltung (siehe Abb. 6.5 auf Seite 142) einen Kurs aus und beginnt mit dessen Ausführung. Angezeigt werden nur jene Kursinhalte, welche auch vom jeweiligen Tutor freigeschalten worden sind (siehe Abb. 6.4 auf Seite 141). Die Navigationselemente entsprechen den vom

Tutor festgelegten Sequencing-Attributen.

2. Tritt während des Ausführens eines Kurses ein Problem bzw. Unklarheit zu einer Lektion auf, besteht für den Studenten nun die Möglichkeit, Hilfestellung beim Tutor zu finden. Dabei kann zur jeweiligen Lektion eine neue Nachricht mit der entsprechenden Problemstellung verfasst werden (siehe Abb. 6.11 auf Seite 148).

   Der Student kann bei der Nachrichtenerstellung bestimmen, ob er nur mit dem Tutor kommunizieren möchte oder ob seine Fragestellung allen Kursteilnehmern ersichtlich sein soll.

3. Nach Absenden der Frage erscheint diese im Diskussionsforum (siehe Abb. 6.12 auf Seite 149) und ein Tutor kann nun helfend eingreifen:

   Eine Unterstützung ist sowohl durch unterstützende Antworten und Hinweise im MessageBoard als auch mittels Freischaltung zusätzlicher Kurslektionen möglich. Hierfür ist es zunächst notwendig, entsprechende Lektionen mit dem integrierten Authoring-Tool zu erstellen (siehe Abb. 6.7 auf Seite 144). Innerhalb der Editier-Ansicht eines Kurses bestimmt der Tutor welche Lektionen für den Frage stellenden Studenten freigeschalten werden und verknüpft diese mit der verursachenden Nachricht (siehe Abb. 6.11 auf Seite 148). Dieser ganze Ablauf wird als Aktion im System gespeichert (Abb. 6.9 auf Seite 146).

   Bei Beantwortung der Fragen innerhalb des MessageBoards obliegt es dem Tutor zu bestimmen, wer die Nachrichten sehen darf.

4. Innerhalb des Diskussionsforums erhält der Student nun eine Nachricht, dass zusätzliche Lektionen für ihn freigeschalten wurden. Der Student führt dann die zusätzliche Lektion aus.

Dieses Szenario lässt sich beliebig oft fortsetzten, ein Tutor kann jeweils individuell Lernpfade einzelner Studenten (oder auch mehrerer gleichzeitig) anpassen, indem er Lektionen hinzufügt oder ausblendet. Weiters kann er mit Hilfestellungen innerhalb des Forums oder Kurses weitere Unterstützung bieten.

Der hier gezeigte Ansatz macht schnell deutlich, dass eine mögliche Automatisierung des Vorganges wesentlich in einer Steigerung der Effizienz beitragen würde. Eine „manuelle“ Betreuung der Studenten ist ab einer gewissen Teilnehmerzahl nicht mehr sinnvoll.

Bereits nachdem ein Student eine Frage gestellt hat, wäre es sinnvoll, nach ähnlichen Fragestellungen innerhalb des Forums und damit verbundenen Aktionen zu suchen und die möglichen Ergebnisse an den Studenten weiterzuleiten. Der Student kann dann immer noch selbst entscheiden, ob die automatisch gefundenen Lösungsvorschläge hilfreich sind oder ob er dennoch Fragen an den Tutor richten möchte.

Konzepte einer Adaptierung des Simple SCORM Players hinsichtlich einer Klassifizierung und automatischen Auffinden ähnlicher Nachrichten in der Arbeit von Thomas Knall ([Kna05]) ausführlichst diskutiert und implementiert – Abb. 6.13 zeigt das Ergebnis. Auf eine weitere Erläuterung dieses Konzeptes wird daher verzichtet.

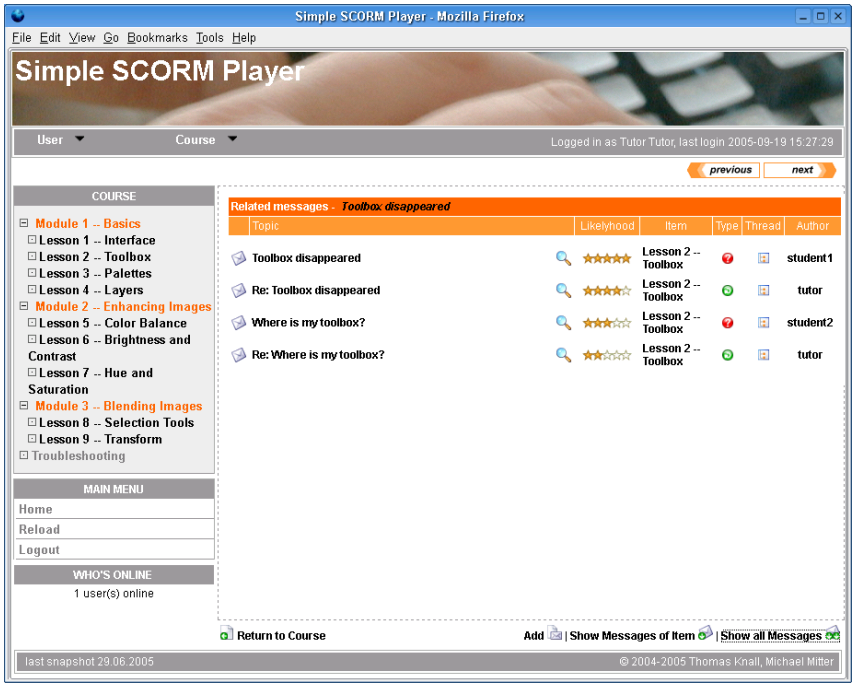


Abb. 6.13: Anzeige ähnlicher Nachrichten

6.4 Details zur Implementierung

Die Grundlage des Simple SCORM Players bilden der Servlet-Container Tomcat[2], der Apache[3] Webserver, MySQL[4] und das Struts[5] Framework. Unterstützte Versionen der verwendeten Software und Bibliotheken sind in Anhang A vorzufinden. Weiters findet sich in Anhang B eine genaue Installations- und Konfigurationsbeschreibung.

An dieser Stelle wird nochmals versucht, überblicksmäßig die Architektur des Simple SCORM Players darzustellen. Generell wurden bereits die grundlegenden Details zur Implementierung und das Zusammenspiel Struts-spezifischer Komponenten in Abschnitt 5.1 bis 5.4 auf den Seiten 112–133 genannt.

6.4.1 Struts Framework

Die Architektur der Anwendung ist durch die Verwendung von Struts und damit verbunden des MVC-Prinzips vorgegeben. Sämtliche Module, die direkt in Zusammenhang mit der Web-Applikation stehen, folgen daher fast immer dem gleichen Aufbau:
View-Komponenten bestehen aus JSP-Dateien, in denen neben üblichen HTML-Tags ausschließlich die Struts-EL- und JSTL- Tag-Bibliotheken verwendet werden. Der komplette Java-Code wurde damit aus der View verbannt (siehe Abschnitt 5.3.5.1 bis 5.3.5.2 auf den Seiten 127–128).

ActionForms werden durch DynaValidationForms bzw. DynaActionForms mit entsprechenden Definitionen innerhalb der Struts-Konfigurationsdatei ersetzt (siehe Abschnitt 5.3.2 auf Seite 123). Für die Validierung der Formulareingaben wird der integrierte Validator in modifizierter Form verwendet (siehe Abschnitt 5.3.1 auf Seite 122). Datenbankabfragen erfolgen innerhalb von Java-Beans unter Verwendnung von Prepared-Statements und Connection-Pooling (siehe Abschnitt 5.3.7.2 bis 5.3.7.3 auf den Seiten 131–133). Weiters werden sämtliche setter- und getter-Methoden in DTOs ausgelagert (siehe Abschnitt 5.3.3 auf Seite 125).

[2] http://jakarta.apache.org/tomcat/
[3] http://httpd.apache.org
[4] http://www.mysql.com
[5] http://struts.apache.org

Die Steuerung und das ActionMapping erfolgt üblicherweise innerhalb einer Action-Klasse. In Abb. 6.14 wird der Zusammenhang und Aufbau einer typischen „DTO-Java-Bean-Action“-Struktur vereinfacht skizziert.
Typische Struts-Merkmale wie Internationalisierung mittels Properties-Dateien wurden ebenfalls implementiert.

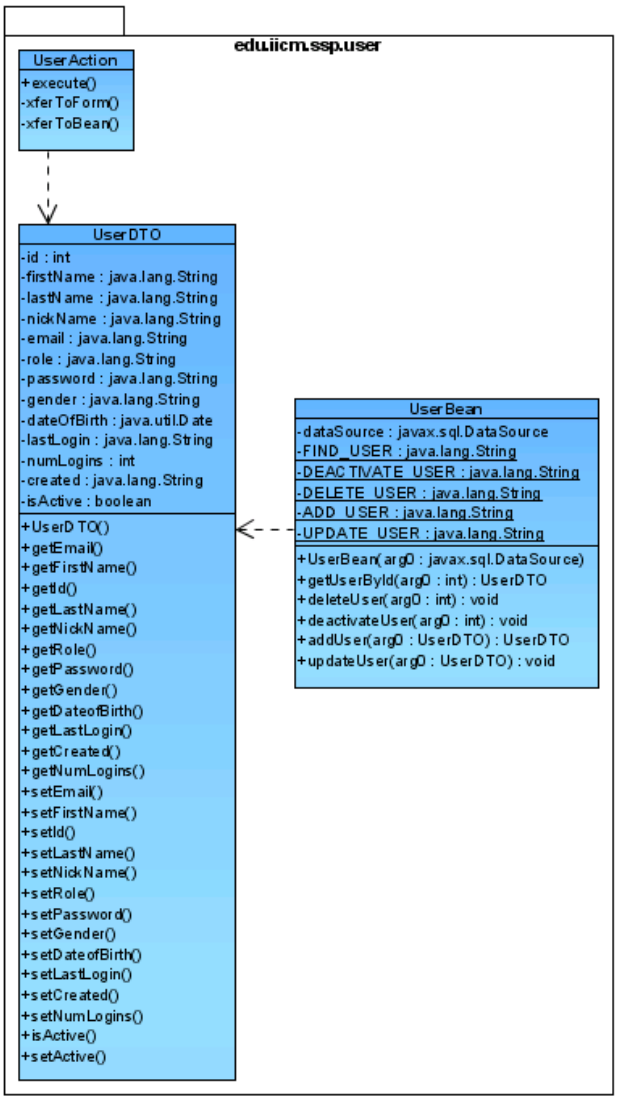


Abb. 6.14: UML Diagramm - DTO, Java-Bean, Action

6.4.2 Paket-Hierarchie

Das Simple SCORM Player Java-Package unterteilt sich in mehrere Sub-Packages. Sämtliche für die Web-Applikation wichtigen Klassen sind im Paket `edu.iicm.ssp.*` enthalten. Weitere Pakete umfassen den modifizierten Parser und Validator von ADL sowie Klassen für die Klassifizierung.

Java-Package	Beschreibung
`edu.iicm.ssp.login`	Anmeldung mit Zusatzfunktionen
`edu.iicm.ssp.user`	Benutzerverwaltung
`edu.iicm.ssp.course`	Kursverwaltung
`edu.iicm.ssp.course.input`	Kursimport
`edu.iicm.ssp.course.item`	Manipulation von Kurselementen
`edu.iicm.ssp.messageboard`	Diskussionsforum
`edu.iicm.ssp.messageboard.message`	Manipulation von Nachrichten
`edu.iicm.ssp.navigation`	Navigationsereignisse
`edu.iicm.ssp.navigation.sequencing`	Sequencing-Steuerung
`edu.iicm.ssp.adaption`	Ähnlichkeitssuche
`edu.iicm.ssp.adaption.activity`	Tutor-Aktivitäten
`edu.iicm.ssp.adaption.classification`	Einbinden von Klassifikator
`edu.iicm.ssp.adaption.similarity`	Schnittstellen für Klassifikator
`edu.iicm.ssp.helper`	Nützliche Tools

Tab. 6.1: Pakete des Simple SCORM Player

Neben den beschriebenen Modulen existieren noch eine Vielzahl weiterer, kleinerer Module und Java-Klassen. Es wurden Modifikationen und Erweiterungen hinsichtlich Struts und Apache Tomcat vorgenommen; Beispiele hierfür sind Änderungen im Requestprozessor und ExceptionHandler, Erweiterungen innerhalb des Validators oder die Integration von log4j. Weiters existieren innerhalb der Packages noch weitere, generische Klassen. Ein genaue Kenntnis dieser Erweiterungen ist jedoch für das grundlegende Verständnis des Simple SCORM Players nicht zwingend erforderlich.

Abb. 6.15 auf der folgenden Seite zeigt überblicksmäßig die Package-Struktur der Applikation.

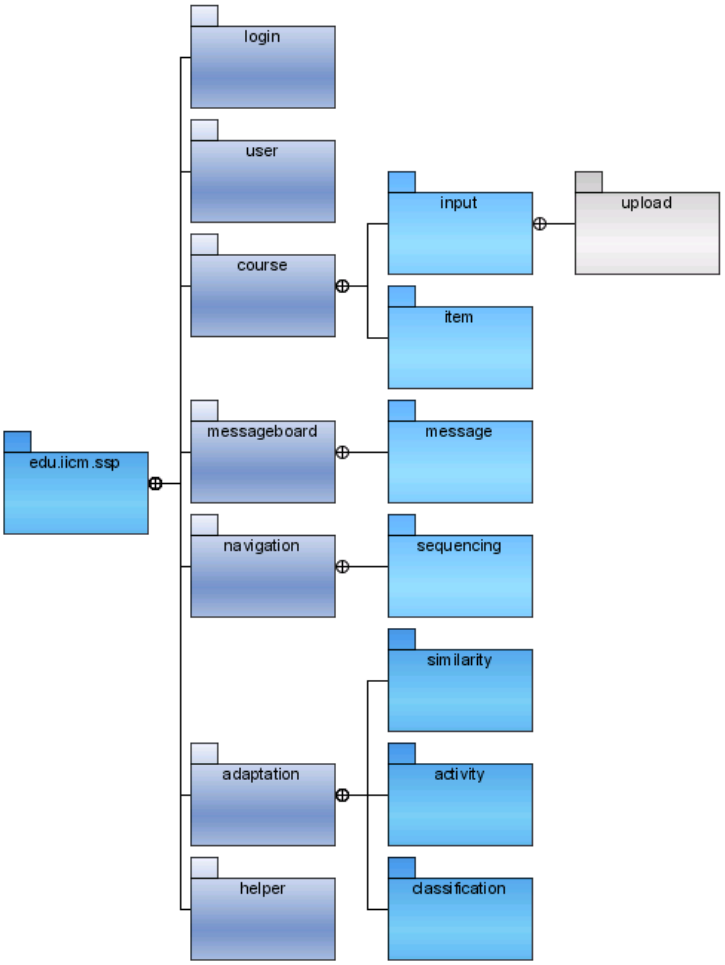


Abb. 6.15: Prinzipielle Paket-Hierarchie des Simple SCORM Players

6.4.3 Datenbank-Struktur

Sämtliche dynamischen Daten des Simple SCORM Player liegen in Form von Einträgen in Datenbanktabellen vor. Als relationale Datenbank kommt MySQL ab Version 4.1 in Verwendung und ersetzt die von ADL vorgesehene MS Access Datenbank.

Aus Performance- und Sicherheitsgründen wurden in allen Klassen SQL-Statements zur Gänze mittels Prepared Statements (siehe Abschnitt 5.3.7.3 auf Seite 133) implementiert. Einen weiteren Geschwindigkeitsgewinn bot die Verwendung von Connection-Pooling (Abschnitt 5.3.7.2 auf Seite 131).

Die Datenbank-Struktur der Applikation lässt sich grob in zwei Bereiche unterteilen: Die Tabellen `application`, `courseInfo` und `itemInfo` sind für den Importvorgang notwendig. Es sind dabei sämtliche geparste Attribute eines importierten Kurses innerhalb dieser Tabellen verfügbar. Die Namensgebung der Tabellen und dessen Spalten entsprechen jener, die ADL auch innerhalb der MS Access Datenbank verwendet hat. Die weiteren Tabellen (`users`, `tree`, `itemMask`, `files`, `activities` und `qac`) sind für einzelne Module der Web-Applikation erforderlich.

In Tabelle 6.2 erfolgt eine Auflistung aller Tabellen und in Abb. 6.16 auf der nächsten Seite wird in stark vereinfachter Form die hier genannte Datenbankstruktur dargestellt.

Tabelle	Beschreibung
`applicationData`	Attribute für Importvorgang
`courseInfo`	grundlegende Kursdaten und Sequencing-Informationen
`itemInfo`	sequentielle Auflistung alle verfügbaren Items, sortiert nach Kurszugehörigkeit und Item-ID
`users`	Benutzer-Attribute
`tree`	Einträge für das Struts-Menü
`itemMask`	Benutzer-bezogene, ausgeblendete Kurseinheiten
`files`	referenzierte Dateien für Authoring
`activities`	Tutoraktionen
`qac`	Nachrichten innerhalb des MessageBoards

Tab. 6.2: MySQL-Tabellen des Simple SCORM Players

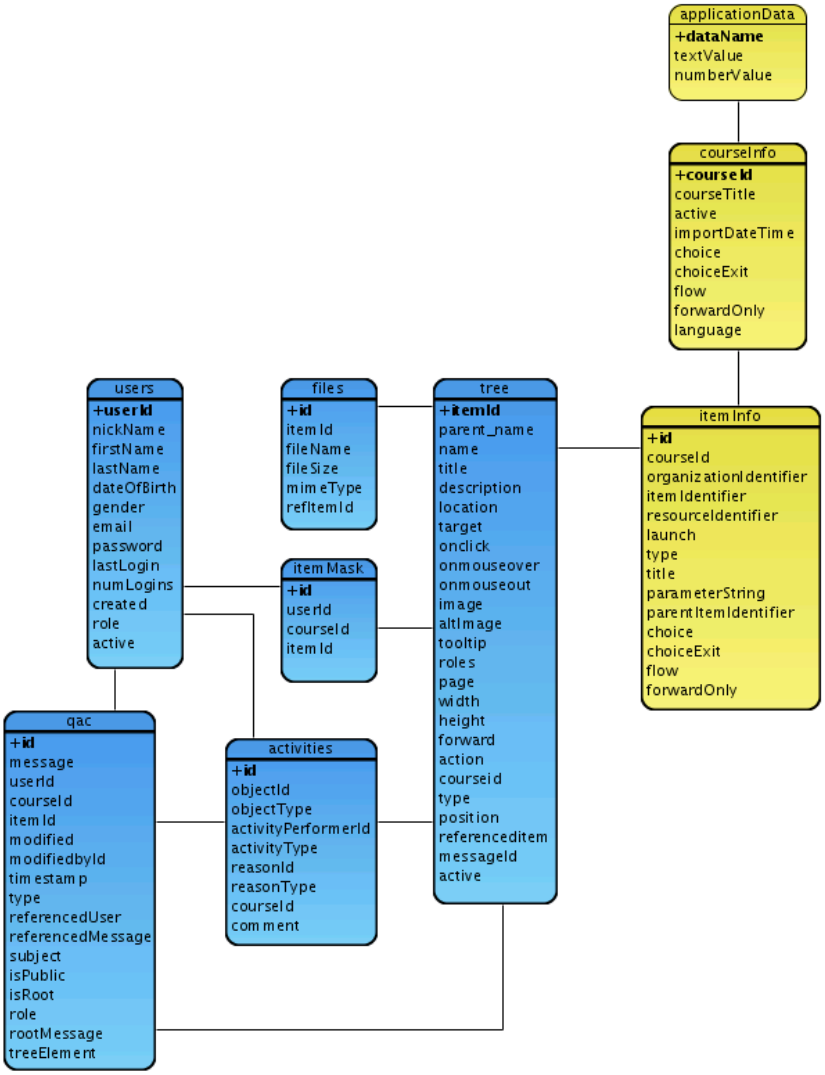


Abb. 6.16: Vereinfachtes Datenbank-Schema

6.4.4 SCORM Integration

6.4.4.1 Import von SCORM-Kursen

Vorrangiges Ziel bei der Implementierung des Prototypen war auch eine Unterstützung von SCORM-konformen Kursinhalten. Zwingende Voraussetzung dafür ist ein Parsen und Validieren der ZIP-Packages während eines Importvorgangs. Hierfür stellt ADL innerhalb des „ADL SCORM 1.3 Sample Run-Time Environment“ sämtlichen Sourcecode des ADL Validators und Parsers zur freien Verfügung.

Im Simple SCORM Player findet sich der ADL-Parser im Modul `org.adl.*` des SSP in angepasster Form wieder. Die ADL-Implementierung wurde dahingehend adaptiert, dass die verwendete Microsoft Access Datenbankanbindung durch eine MySQL-Datenbankanbindung ersetzt wurde. Weiters wurde die Speicherung der Daten entsprechend der Web-Applikation angepasst.
Auch hier werden SQL-Anweisungen unter Verwendung von PreparedStatements ausgeführt, um so einen Performance- und Sicherheitsgewinn zu erhalten. Nichtverwendete Klassen des Parsers wurden entfernt, dies führte zu einer übersichtlicheren Struktur und auch zu einem Geschwindigkeitsgewinn.

Importvorgang

Bei Laden eines Kurses via der Web-Applikation wird das SCORM-Paket entsprechend geparst und validiert. Das ZIP-File muss dabei einem gültigem „SCORM 2004 Content Aggregation Package“ entsprechen. Nach erfolgreicher Validierung wird das Paket in einen entsprechenden Ordner innerhalb des SSP entpackt. Sämtliche Attribute der geparsten Dateien werden in entsprechende Tabellen der Datenbankstruktur geschrieben und stehen nun dem System zur Verfügung. Der importierte Kurs ist innerhalb der Kursverwaltung (siehe Abschnitt 6.2.3 auf Seite 140) sichtbar.

Nachdem der Validierungsvorgang recht aufwendig und oftmals bei großsen Kurspaketen viel Zeit in Anspruch nimmt, erweist es sich als durchaus sinnvoll, bei Kursen, wo eine SCORM-Konformität bereits im Vorfeld bekannt ist, den Validator zu deaktivieren. Die vorzunehmenden Einstellungen hinsichtlich Pfadangaben und Validator sind im Konfigurationsfile des Simple SCORM Player vorzunehmen.

Details hierfür sind in Unterabschnitt B.1.3 zu entnehmen.

Neben eines Imports mittels Web-Interface besteht für versierte User auch die Möglichkeit, via einer Shell, Kurs-Pakete zu importieren. Hierfür stehen innerhalb des Simple SCORM Player Shell-Tools im Paket `edu.iicm.run` zur Verfügung.

Grundsätzlich ist anzumerken, dass der von ADL zur Verfügung gestellte Source Code schlecht dokumentiert und strukturiert ist. Oftmals entstand während der Implementierungsphase der Eindruck, dass es seitens ADL keine klaren Implementierungsrichtlinien gab.

6.4.4.2 Sequencing

Im Zuge der Entwicklung der Web-Applikation war zumindest eine einfache Form des Sequencing zu implementieren. Nach eingehender Analyse und Evaluierung wurden Teile der so genannten *Sequencing Control Modes* (siehe Abschnitt 3.3.2.1 auf Seite 63) ausgewählt. Wie bereits in der theoretischen Abhandlung erläutert, dienen diese zur Spezifizierung von Elementen in Navigationsmenüs angewandt auf einen Cluster – einer Verschachtelung von Kurs-Items. Mittels Setzen entsprechender Werte lässt sich beispielsweise bestimmen, welche Elemente innerhalb einer Menüstruktur direkt ausgewählt werden dürfen und welche nur statisch sind.

Implementiert wurden die Modi *Sequencing Control Choice* (siehe Abb. 3.15 auf Seite 64) und *Sequencing Control Flow* (siehe Abb. 3.16 auf Seite 64). Der erste Modus ermöglicht eine beliebige Navigation innerhalb eines Kurses, zweiterer beschränkt sich auf ein rein sequentielles Navigieren. Beide Modi können auch in kombinierter Form angewandt werden.

Bereits zu Beginn hat jeder Kurs entsprechende Default-Werte, die während des Importvorganges geparst und in die entsprechende Tabelle geschrieben werden. Im Simple SCORM Player entsprechen Sequencing-Informationen einzelnen Einträgen innerhalb der Datenbank-Tabelle `courseInfo`. Ein Modifizieren erfolgt seitens des Tutors mittels entsprechenden Editiermöglichkeiten innerhalb eines Kurses (siehe Abb. 6.17 auf der folgenden Seite). Die für das Sequencing notwendigen Navigationselemente werden mittels einfachen Vor- und Zurück-Buttons (siehe Abb. 6.19 auf der nächsten Seite) und unter Verwendung des Struts-Menus rea-

lisiert (siehe Abschnitt 5.3.4 auf Seite 126 und Abb. 6.18).

Abb. 6.17: Festlegen von Sequencing-Attributen

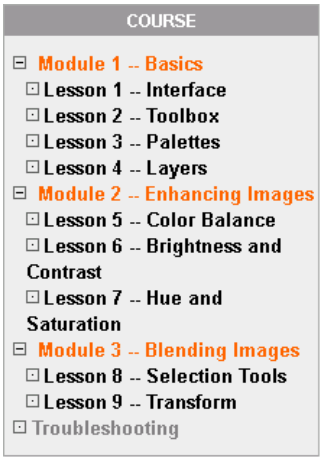

COURSE
Module 1 -- Basics
Lesson 1 -- Interface
Lesson 2 -- Toolbox
Lesson 3 -- Palettes
Lesson 4 -- Layers
Module 2 -- Enhancing Images
Lesson 5 -- Color Balance
Lesson 6 -- Brightness and Contrast
Lesson 7 -- Hue and Saturation
Module 3 -- Blending Images
Lesson 8 -- Selection Tools
Lesson 9 -- Transform
Troubleshooting

Abb. 6.18: Menüstruktur (Struts-Menu)

Abb. 6.19: Navigationselemente

6.4.4.3 Einschränkungen

Bereits bei der Analyse der SCORM-Spezifikation in Kapitel 3 auf Seite 34 wurde festgestellt, dass eine clientseitige Implementierung des API-Adapters erhebliche Probleme und auch Nachteile mit sich führt. In der Spezifikation werden sämtlichen Funktionen der API mittels JavaScript-Funktionen implementiert. Neben einem Aufrufen verschiedener Plugins wie zum Beispiel Flash-Filmen, wird mit diesen Funktionen auch die Auswertung von verschiedenen Tests hinsichtlich erreichter Punkteanzahl oder Überprüfung eines Zeitlimits realisiert.

Bei der Entwicklung des Simple SCORM Player wurden aufgrund diese Unzulänglichkeiten und offensichtlichen Nachteile – auch in Verbindung mit Struts – von den Autoren beschlossen, auf eine Unterstützung dieser JavaScript-Funktionen zu verzichten und sich vermehrt dem ADL Parser und Validator zu widmen.

Kapitel 7

Zusammenfassung und Ausblick

Das primäre Ziel dieser Magisterarbeit bestand in der Implementierung eines prototypischen Lern-Management-Systems unter besonderer Berücksichtigung der SCORM Spezifikationen. Weiters sollten Konzepte gefunden werden, effiziente Tutoring-Möglichkeiten zu integrieren. Während im Untersuchungsbereich theoretische Grundlagen diskutiert wurden, umfasste der Gestaltungsbereich die Implementation eines beispielhaften Lernsystems.

Für den Einstieg in die Thematik wurden grundlegende Begrifflichkeiten hinsichtlich E-Learning definiert und erläutert. Beginnend mit einfachen, elektronischen Lernsystemen wurden auch komplexere Konzepte wie Intelligentes Tutoring umrissen. Weiters erfolgte eine Vorstellung aktueller Standardisierungsgremien und deren veröffentlichten Standards und Spezifikationen.

Das zu implentierende Learning-Management-System sollte zumindest einen Standard unterstützen – Aufgrund guter Dokumentation und breiter Unterstützung fiel die Wahl auf SCORM (Sharable Content Object Reference Model). Somit war es zwingend erforderlich, die E-Learning-Spezifikation in detaillierter Form zu untersuchen. Dabei wurde neben Architektur und Struktur typischer Kurseinheiten auch besonderes Augenmerk auf die neu hinzugekommene Ablaufsteuerung gelegt. Abschließend erfolgte ein Ausblick auf zukünftige Entwicklungen hinsichtlich der Spezifikation.

Für die Implementation des LMS war es auch erforderlich, zunächst grundlegende Eigenschaften eines typischen Systems zu erläutern. Hierfür wurden allgemeine Anforderungen diskutiert und im speziellen am Markt verfügbare Learning-

Management-Systeme evaluiert. Es fand dabei eine Untersuchung hinsichtlich Funktionalität, Vor- und Nachteile, Architektur und einer Unterstützung von Standards statt. Für die Evaluierung wurden Produkte sowohl aus dem kommerziellen als auch dem Open Source Bereich ausgewählt.

Den theoretischen Grundlagen folgte die Beschreibung der implementierten Applikation. Zunächst wurde das für die Anwendung verwendete Java-Framework Struts hinsichtlich seiner Architektur und Möglichkeiten betrachet und analysiert. Darauf folgend wurden sinnvolle Modifikationen und Erweiterungen des Frameworks veranschaulicht.

Schließlich wurden die elementarsten Module der Web-Applikation „Simple SCORM Player“ vorgestellt. Bei der Funktionalität der Anwendung wurde versucht, die Basisanforderungen eines typischen Lern-Management-Systems zu integrieren – beginnend mit asynchronen Kommunikationsmöglichkeiten bis hin zu einem einfachen Authoring von Kursinhalten. Zusätzlich wurde kurz veranschaulicht, wie eine Unterstützung von Studenten bei Ausführen von Kursinhalten innerhalb der Web-Anwendung ermöglicht wird bzw. wurde auf weitere Konzepte hingewiesen.

Während der Implementierung der Applikation traten auch vereinzelt Problemstellungen und Grenzen hinsichtlich der Integration von SCORM auf. Für ein Importieren von fertigen SCORM-Kursen musste der von ADL implementierte Parser erst entsprechend an die Web-Anwendung angepasst werden. Weiters führte die Unterstützung der Sequencing and Navigation-Spezifikation aufgrund der clientseitigen Kommunikation via API-Adapter zu Einschränkungen. Bei der Analyse hinsichtlich studentischer Unterstützungmöglichkeiten mittels Lernpfaden und Authoring zeigte sich rasch, dass eine Automatisierung des Vorgangs unumgänglich ist.

Positiv anzumerken ist, dass eine Nutzung von Standards in Zusammenhang mit E-Learning und hier besonders bei der Entwicklung von Lerninhalten und Lernumgebungen aufgrund erhöhter Interoperabilität und Wiederverwendbarkeit durchaus sinnvoll ist. SCORM erfreut sich bereits einer breiten Unterstützung innerhalb der Entwicklergemeinde und hat durchaus Potential, sich als *der* E-Learning-Standard zu etablieren. Vor allem ausreichend zur Verfügung stehende Dokumentationen und Werkzeuge unterstützen dies. Mit dem neu eingeführten Se-

quencing and Navigation besteht die Möglichkeit, ein komplexes Kursdesign mit anspruchsvoller Ablaufsteuerung zu implementieren.

Dennoch stellt E-Learning trotz der teilweise anfänglichen Euphorie und technischen Fortschritte keineswegs ein Allheilmittel für zukünftige Wissensvermittlung dar – mittlerweile herrscht eher eine realistische Chanceneinschätzung. Derzeitigen Lernumgebungen mangelt es noch an den Möglichkeiten, unterschiedliche didaktische Lernszenarien zu implementieren, auch besteht hier ein Mangel an Flexibilität. Ein hoher technischer Funktionsumfang reicht nicht aus, um dies zu kompensieren.

Die Zukunft von E-Learning liegt laut Expertenmeinungen ([TZ03]) daher in einer Verwendung mehrerer, unterschiedlicher Lernformen, dem so genannten *Blended Learning* – eine sinnvolle Kombination von plattformbasierendem Lernen und Präsenzunterricht. Diese Entwicklung hat zur Folge, dass auch beim Design zukünftiger Lernumgebungen die Vorteile unterschiedlicher Lernformen genutzt und implementiert werden müssen. Weiters werden sich auch die Rollen und Aufgabenverteilung von Lehrenden ändern. Lehrer müssen in der Lage sein, E-Learning so einzusetzen, dass dieses durch den Präsenzunterricht sinnvoll ergänzt wird und eine individuelle Betreuung der Lernenden ermöglicht. Ein Tutor muss neben fachlichen und didaktischen Kompetenzen, zusätzlich auch technische besitzen. Lernende müssen sich ebenfalls auf die veränderten Anforderungen einstellen – erweiterte technische Kompetenzen und Umgang mit einem PC sind dabei Grundvoraussetzungen, um überhaupt an E-Learning-Prozessen teilnehmen zu können.

Grundsätzlich sind viele der Meinung, dass es zukünftig E-Learning in der heutigen Form nicht mehr geben wird. Ein Großteil der jetzt in Lernumgebungen implementierten Funktionalität wird in das Internet integriert werden, wodurch lediglich ein Browser für einen Zugang zu den Lernressourcen benötigt werden wird.

Anhang A

Verwendete Software

Dieser Anhang dokumentiert die für den Simple SCORM Player (SSP) eingesetzte Software. Diese Liste stellt nicht zwingenderweise eine Voraussetzung zur Nutzung der Web-Applikation dar – diese wäre genauso unter Windows oder mit einem anderen Web-Server lauffähig. Die genauen Voraussetzungen für den SSP sind in Abschnitt B.1.1 auf Seite 171 dokumentiert.

Die verwendete Software teilt sich zum einen in Systemsoftware bei der es sich um grundlegende – für Web-Anwendungen üblicherweise erforderliche – Software handelt und zum anderen in Software die innerhalb der Laufzeitumgebung eingesetzt wird. Letztere muss nicht explizit zu Verfügung gestellt werden, da die hier angeführten Pakete bereits im Installationspaket enthalten sind. Mit der Installation (siehe Abschnitt B.1.2 auf Seite 171) werden diese dann in die entsprechenden Verzeichnisse übertragen.

A.1 Systemsoftware

- Betriebssystem
  - Kubuntu Linux Version 5.04
    `http://www.kubuntu.org`
- Web-Server & Servlet Container
  - Apache HTTP Server Version 2.0.53 mit mod_jk2 Version 2.0.4
    `http://httpd.apache.org`

  - Apache Tomcat Version 5.5.9
    http://jakarta.apache.org/tomcat

- Datenbank-Software
  - MySQL Database Server Version 4.1.13a
    http://dev.mysql.com/downloads/mysql/4.1.html
  - MySQL Connector/J Version 3.0.16
    http://dev.mysql.com/downloads/connector/j

- Laufzeitumgebung
  - Java 2 Platform Standard Edition (J2SE) Version 1.5.0_04-b05
    http://java.sun.com/j2se/1.5.0

A.2 Pakete innerhalb der Laufzeitumgebung

- Präsentationsschicht
  - Struts Framework Version 1.2.4
    http://struts.apache.org
  - Struts Menu Tag Library Version 2.2
    http://struts-menu.sourceforge.net

- Parser
  - ADL SCORM Sample Run-Time Environment Version 1.3 Beta-3
    http://www.adlnet.org/downloads
  - Xerces2 Java Parser Version 2.6.2
    http://xml.apache.org/xerces2-j
  - JDOM Classes Version 0.9
    http://www.jdom.org

- Klassifikation (Adaptierung)
  - Classifier4J Version 0.6
    http://classifier4j.sourceforge.net
  - Apache Lucene Version 1.4.3
    http://lucene.apache.org/java

- Diverses
  - Log4j Version 1.2.9
    http://logging.apache.org/log4j
  - Apache Commons
    http://jakarta.apache.org/commons
  - JavaMail API Version 1.3.2
    http://java.sun.com/products/javamail

Anhang B

Dokumentation

Dieser Anhang enthält eine Beschreibung der Schritte zur Installation und Konfiguration des Simple SCORM Players wobei grundlegende Kenntnisse über den Apache HTTP Server, Apache Tomcat und MySQL vorausgesetzt werden.

B.1 Einrichtung der Web-Applikation

Für die Einrichtung des Simple SCORM Players wird davon ausgegangen, dass der Prototyp als `SCORMplayer.war` – in Form eines Web Application Archive (WAR) – vorliegt. Das Deployment[1] gliedert sich in vier Schritte:

(1) **Entpacken** – Das WAR muss in den entsprechenden Kontext kopiert und dort entpackt werden.

(2) **MySQL** – Über ein MySQL-Skript wird die Datenbank „scormplayer“ erstellt und initialisiert.

(3) `struts-config.xml` – Benutzername und Passwort für einen Zugang zur MySQL-Datenbank müssen eingetragen werden, um dem SSP die Nutzung der Datenbank zu ermöglichen.

(4) `SCORMplayer.properties` – Angabe eines Verzeichnisses, das später Kurse enthalten wird und eines temporären Verzeichnisses, das zur Zwischenspeicherung bei Importvorgängen dient.

[1] Als Deployment bezeichnet man die Verteilung und Installation von Software einschließlich deren Konfiguration.

B.1.1 Voraussetzungen

Die für diese Applikation verwendete Software inklusive der Versionsnummern wurde bereits in Anhang A genannt. Der Autor kann die uneingeschränkte Funktionsfähigkeit des SSP ausschließlich für diese Versionen garantieren. Neuere Versionen sollten grundsätzlich dennoch kein Problem darstellen.

Bei der Verwendung älterer Komponenten sollte jedoch folgendes berücksichtigt werden:

- Der MySQL Database Server muss mindestens die Version 4.1.x aufweisen. Erst diese Version erlaubt die Verwendung spezieller Abfragen, die beim Simple SCORM Player zum Einsatz kommen.
- Die aktuelle Version 5.5.9 von Apache Tomcat setzt eine Java-Version 1.5.x voraus. Soll eine ältere Java-Version zum Einsatz kommen, kann maximal Apache Tomcat Version 5.0.x verwendet werden.

B.1.2 Installation

Die Installation beschränkt sich im grossen und ganzen auf das Entpacken der WAR-Datei. Dies kann auf zwei verschiedene Arten durchgeführt werden.

Die klassische Methode wird über eine Shell ausgeführt:

(1) Apache Tomcat stoppen.

(2) Ein existierendes Deployment löschen – Sollte sich eine frühere Installation im Verzeichnis `$CATALINA_HOME/webapps/SCORMplayer/`[2] befinden, muss dieses mitsamt seinem Inhalt gelöscht werden.

z. B. mit `rm -r $CATALINA_HOME/webapps/SCORMplayer`

(3) `SCORMplayer.war` in das Verzeichnis `$CATALINA_HOME/webapps/` kopieren.

(4) Apache Tomcat wieder starten. Die WAR-Datei wird nun automatisch entpackt und der Kontext gestartet.

[2] `$CATALINA_HOME` bezeichnet das Installationsverzeichnis von Apache Tomcat, beispielsweise „`/usr/local/tomcat/`“. Ältere Tomcat-Versionen bezeichnen dies als `$TOMCAT_HOME`.

Die zweite – komfortablere Möglichkeit – bietet Apache Tomcat ab der Version 4.x durch die Nutzung einer speziellen Web-Applikation, dem „Tomcat Webanwendungs-Manager“ der über die URL `http://host:8080/manager/html` aufgerufen werden kann. Dabei wird das WAR über ein Formular in den entsprechenden Kontext geladen, dort entpackt und schließlich als neuer Kontext `SCORMplayer` gestartet.

MySQL-Skript `$CATALINA_HOME/webapps/SCORMplayer/sql-script/`

Nun muss die MySQL-Datenbank für den Simple SCORM Player vorbereitet werden. Dafür ist der Import des SQL-Skripts `prepare_table_scormplayer.sql` vorgesehen. Sollte dem Leser nicht bekannt sein, wie SQL-Skripte ausgeführt werden, sei hier auf den Abschnitt „MySQL im Stapelbetrieb (Batch Mode)[3]“ der MySQL-Dokumentation verwiesen. Dieses Skript erstellt die Datenbank „scormplayer“ zusammen mit einigen Tabellen, die mit Default-Werten initialisiert werden.

B.1.3 Konfiguration

Nach der Installation des Simple SCORM Players müssen zumindest die Zugangsdaten zum MySQL-Server und einige Pfade bekanntgegeben werden um einen fehlerfreien Betrieb des SSP zu gewährleisten.

Hinweis zu Pfad-Angaben: Der Leser möge bei der Angabe von Pfaden die Trennzeichen des eingesetzten Betriebssystems beachten. Unter Linux wird „/“ verwendet während unter Windows „\“ zum Einsatz kommt, das jedoch als „\\“ eingetragen werden muss. Um eine einheitliche Schreibweise zu gewährleisten wird in dieser Arbeit stets „/“ als Trennzeichen verwendet. Windows-Benutzer mögen dies berücksichtigen. Das abschließende „/“ bzw. „\\“ jeder Pfadangabe ist optional.

struts-config.xml `$CATALINA_HOME/webapps/SCORMplayer/WEB-INF/`

Listing B.1 auf der nächsten Seite zeigt den typischen Kopf einer SSP-Konfigurationsdatei, der `struts-config.xml`. Für den Leser sind ausschließlich die Zeilen 6 bis 8 auf der folgenden Seite relevant. Diese müssen einen gültigen MySQL-Benutzernamen und das dazugehörende Passwort enthalten. Zeile 8 legt schließ-

[3] `http://dev.mysql.com/doc/mysql/de/batch-mode.html`

lich die URL zum Datenbankserver mitsamt der Datenbank „scormplayer“ fest.

Hinweis: Datenbank-Root-Rechte sind zur Nutzung des SSP nicht erforderlich.

```
<struts-config>
  <data-sources>
    <data-source key="scorm" type="org.apache.commons.dbcp.
       BasicDataSource">
      <set-property property="description" value="SCORM Database"/>
      <set-property property="driverClassName" value="com.mysql.jdbc.
         Driver"/>
      <set-property property="username" value="MySQLusername"/>
      <set-property property="password" value="MySQLpassword"/>
      <set-property property="url" value="jdbc:mysql://localhost/
         scormplayer"/>
      <set-property property="maxCount" value="3"/>
      <set-property property="minCount" value="2"/>
    </data-source>
  </data-sources>
</struts-config>
```

Listing B.1: Bekanntgabe der MySQL-Zugangsdaten über `struts-config.xml`

SCORMplayer.properties `$CATALINA_HOME/webapps/SCORMplayer/WEB-INF/classes/`

Eine weitere Konfigurationsdatei, `SCORMplayer.properties`, wird in Listing B.2 auf Seite 176 abgebildet. Diese ist keine XML-Datei sondern enthält im typischen Format von Properties-Dateien[4] ausschließlich „key/value“-Paare.

An dieser Stelle folgt nun ein Überblick über die einzelnen Schlüssel. Die ersten drei Schlüssel sind jene, die nach einer Installation unbedingt angepasst werden müssen. Erfolgt dies nicht, können keine Kurs importiert werden.

`scormplayer.path.courses.absolute` – bezeichnet den Pfad zu einem Verzeichnis, das die importierten Kurse enthält. Dabei ist unbedingt zu beachten, dass sich dieses innerhalb des SCORMplayer-Kontexts `$CATALINA_HOME/webapps/SCORMplayer/` befinden muss.

[4] Dabei handelt es sich um Plaintext-Dateien, die als einfacher Konfigurationsmechanismus verwendet werden indem Texte („values“) unter bestimmten Namen („keys“) registriert werden.

`scormplayer.path.courses.relative` – Dieser Pfad entspricht dem beim vorherigen Schlüssel gewählten Pfad relativ zum Kontext. Ist der absolute Pfad beispielsweise `$CATALINA_HOME/webapps/SCORMplayer/foo/`, dann muss als relativer Pfad `/foo` eingetragen werden.

`scormplayer.path.temp` – Legt ein temporäres Verzeichnis fest, das von den ADL-Parser-Klassen während eines Kurs-Importvorgangs als Zwischenspeicher verwendet wird. Dieses Verzeichnis kann frei gewählt werden – solange Apache Tomcat dafür entsprechende Schreibrechte besitzt.

`scormplayer.parser.validating` – Hiermit kann festgelegt werden, dass eine Kurs-Validierung während eines Imports vorgenommen wird (`true`). Erst dadurch ist sichergestellt, dass der Kurs SCORM-konform ist. Um den Vorgang zu beschleunigen kann auf eine Validierung verzichtet werden (`false`). (Default: `true`)

Die bisher erfolgte Konfiguration ist ausreichend, um den Simple SCORM Player in Betrieb zu nehmen. Zu diesem Zweck muss der Kontext – beispielsweise mit dem „Tomcat Webanwendungs-Manager“ – neu gestartet werden.

Die folgenden fünf Schlüssel betreffen die Funktion „Wiederbeschaffung vergessener Passwörter“:

`scormplayer.mail.from` – Solch ein Schlüssel enthält üblicherweise die E-Mail-Adresse des Administrators. Im Fall des Simple SCORM Players entspricht dies der E-Mail-Adresse des Tutors, der in der Funktion eines Administrators agiert.

`scormplayer.mail.subject` – Bezeichnet den Betreff einer automatisiert generierten E-Mail.

`scormplayer.mail.host` – Hier muss ein gültiger SMTP-Server angegeben werden.

`scormplayer.mail.username` – Dies entspricht einem gültigen Benutzernamen für den oben angegebenen Server.

`scormplayer.mail.password` – Das zum oben angegebenen Benutzernamen passende Passwort.

Die nächsten fünf Schlüssel konfigurieren den Klassifikator des Projekts

„Classifier4J“:

`scormplayer.messages.similarity.limit` – Beschränkt die Anzahl der angezeigten ähnlichen Beiträge auf eine bestimmte Zahl (wobei die besten Treffer ausgewählt werden). Dies ist dann sinnvoll, wenn bereits sehr viele Beiträge zu einem Kurs existieren. Der Wert `0` steht dabei für „kein Limit“. (Default: `0`)

`scormplayer.messages.similarity.thread` – Hiermit kann das Training im Zuge einer Klassifikation mit dem gesamten Thread (`true`) oder mit nur einem Beitrag (`false`) bestimmt werden. (Default: `true`)

`scormplayer.messages.similarity.threshold` – Stellt die Festlegung eines Schwellwertes dar. Dieser Wert bestimmt eine Mindestähnlichkeit, die ähnliche Nachrichten aufweisen müssen, um als solche angezeigt zu werden. Im Zuge der Entwicklung des Simple SCORM Players hat sich ein Wert von `0.4` als brauchbar erwiesen. Bei `0` wird keine Nachricht gefiltert und alle Nachrichten als „ähnlich“ präsentiert. (Default: `0`)

`scormplayer.messages.timelimit` – Neue Beiträge werden durch ein Symbol als solche gekennzeichnet. Dieser Schlüssel gibt dafür das maximale Alter „neuer Beiträge“ in Stunden an. (Default: `5`)

`scormplayer.messages.similarity.implementation` – Dieser elementare Schlüssel dient dazu, alternative Algorithmen zu registrieren. Der hier angegebene Klassifikator muss die Schnittstelle `MessageSimilarity` implementieren und kann dazu die Adapterklasse `MessageSimilarityAdapter` verwenden. (Default: `edu.iicm.ssp.adaptation.classification.MessageSimilarityDummy`[5])

Classifier4J.properties `$CATALINA_HOME/webapps/SCORMplayer/WEB-INF/classes/`

Diese Properties-Datei konfiguriert ausschließlich den im Simple SCORM Player implementierten „Vector Space Search“-Algorithmus, genauer gesagt die Klassen des Projekts „Classifier4J“.

Weiters kann mit Hilfe dieser Datei die Entfernung von Stoppworten veranlasst, Stoppwort-Listen sprachabhängig registriert und schließlich eine Wortstammreduktion aktiviert werden.

[5] Hierbei handelt es sich nicht um einen Klassifikator sondern um eine Klasse, die Beiträge nur durchreicht, d. h. keine Filterung vornimmt.

```
1  scormplayer.path.courses.absolute = /usr/local/tomcat/webapps/
     SCORMplayer/courses/
2  scormplayer.path.courses.relative = /courses
3  scormplayer.path.temp = /usr/local/tomcat/webapps/SCORMplayer/temp/

5  scormplayer.parser.validating = true

7  scormplayer.mail.from     = <the admin's e-mail address>
8  scormplayer.mail.subject  = <any subject>
9  scormplayer.mail.host     = <any smtp server>
10 scormplayer.mail.username = <valid username for smtp server>
11 scormplayer.mail.password = <password>

13 scormplayer.messages.similarity.limit     = 0
14 scormplayer.messages.similarity.thread    = false
15 scormplayer.messages.similarity.threshold = 0.4
16 scormplayer.messages.timelimit            = 6
17 scormplayer.messages.similarity.implementation = edu.iicm.ssp.
     adaptation.classification.MessageSimilarityClassifier4J
```

Listing B.2: Konfiguration des SSP über `SCORMplayer.properties`

Hier folgt ebenfalls eine Kurzbeschreibung einzelner Schlüssel, deren Anpassung im Zuge einer Installation jedoch nicht unbedingt erforderlich ist.

`classifier4j.stopwords.enabled` – Aktiviert die Entfernung von Stoppworten (`true`) bzw. verhindert deren Entfernung (`false`). (Default: `false`)

Hinweis: Ist die Stoppwort-Entfernung aktiviert müssen die folgenden beiden Schlüssel ebenfalls konfiguriert werden.

`classifier4j.stopwords.repository.english` – Angabe einer Stoppwort-Liste für Englisch.

`classifier4j.stopwords.repository.german` – Angabe einer Stoppwort-Liste für Deutsch.

`classifier4j.language.default` – Ist für einen Kurs wider Erwarten keine Sprache festgelegt, dann wird die hier angegebene (`german`, `english`) als Ersatz verwendet. (Default: `german`)

`classifier4j.stemming` – Aktiviert (`true`) eine Wortstammreduzierung. (Default: `false`)

```
1 classifier4j.stopwords.enabled = enabled
2 classifier4j.stopwords.repository.english = stopwords.english.txt
3 classifier4j.stopwords.repository.german = stopwords.german.txt
4 classifier4j.language.default = german
5 classifier4j.stemming = enabled
```

Listing B.3: Konfiguration des Klassifikators über `Classifier4J.properties`

Stoppwortlisten `$CATALINA_HOME/webapps/SCORMplayer/WEB-INF/classes/`

Stoppwort-Listen entsprechen einfachen Plaintext-Dateien wobei jede Zeile genau ein Stoppwort enthält.

Hinweis: Eine Sortierung der Worte innerhalb der Listen ist nicht erforderlich.

Zum Beispiel:

```
ab
als
also
am
an
auf
aus
bei
...
```

Achtung: Nach erfolgter Konfiguration ist der Kontext `SCORMplayer` unbedingt – zum Beispiel über den „Tomcat Webanwendungs-Manager“ – neu zu starten.

B.1.4 Neukompilierung – build.xml `$CATALINA_HOME/webapps/SCORMplayer/`

Für den Fall, dass eine Neukompilierung der Java-Quelltexte gewünscht wird, kann die Datei `build.xml` mit Kompilierungsanweisungen für Apache Ant[6] verwendet werden.

[6] http://ant.apache.org

Für die Ausführung mittels Apache Ant stehen folgende zwei Möglichkeiten zur Auswahl:

`ant` – Entspricht der Option `ant all` und bewirkt eine Neukompilierung sämtlicher Quelltexte. Die dadurch erstellten Bytecode-Dateien finden sich dann im Verzeichnis `$CATALINA_HOME/webapps/SCORMplayer/WEB-INF/classes/`. Vor der Kompilierung wird ein `ant clean` aufgerufen (siehe unten) und dadurch alte Bytecode-Dateien gelöscht.

`ant clean` – Löscht alle bisher erstellten Bytecode-Dateien aus dem Verzeichnis `$CATALINA_HOME/webapps/SCORMplayer/WEB-INF/classes/`.

Hinweis: Die für diese Arbeit kompilierten Klassen des Simple SCORM Players wurden zu einem JAR[7] als `SCORMplayer.jar` zusammengefasst im Verzeichnis `$CATALINA_HOME/webapps/SCORMplayer/WEB-INF/lib/` gespeichert.

[7] Dabei handelt es sich um eine ZIP-Datei, die komprimierte Bytecode-Dateien enthält.

Abkürzungsverzeichnis

ADL . . . . . Advanced Distributed Learning 5

AGRs . . . . . AICC Guidelines and Recommendations 26

AICC . . . . . Aviation Industry Computer Based Training Commitee 5

AIX . . . . . Advanced Interactive eXecutive 92

AMP . . . . . Apache-PHP-MySQL 103

ANSI . . . . . American National Standards Institute 25

API . . . . . Application Programming Interface 5

ARIADNE . . . . Alliance of Remote Instructural Authoring and Distribution Networks for Europe 5

ASCII . . . . . American Standard Code for Information Interchange 26

AU . . . . . Assignable Unit 26

BMBWK . . . . . Bundesministerium für Bildung, Wissenschaft und Kultur 81

CAM . . . . . Content Aggregation Model vii

CBI . . . . . Computer Based Instruction 11

CBT . . . . . Computer Based Training 3

CD-ROM . . . . . Compact Disc Read-Only Memory 10

CE . . . . . Campus Edition 94

CEN . . . . . European Committee for Standardization 27

CI . . . . . Corporate Identity 83

CLIX . . . . . Corporate Learning and Information Exchange 92

CMI . . . . . Computer Managed Instruction 29

CMS . . . . . Content Management System 16

CP . . . . . Content Packaging 32

CSCL . . . . . Computer Supported Cooperative Learning 14

CSCW . . . . . Computer Supported Cooperative Work 14

CSS . . . . . Cascading Style Sheets 100

DBMS . . . . . Database Management System 132

DCMES . . . . . Dublin Core Metadata Element Set 28

DCMI . . . . . Dublin Core Meta Initiative 5

DIN . . . . . Deutsches Institut für Normung 25

DoD . . . . . Department of Defense 35

DOM . . . . . Document Object Model 53

DTD . . . . . Document Type Definition 200

DTO . . . . . Dynamic Transfer Object 127

DVD . . . . . Digital Video Disc 14

ECMA . . . . . European Computer Manufacturers Association 37

EL . . . . . Expression Language v

EML . . . . . Educational Modeling Language 28

ERIC/IT . . . . . Education Resources Information Center on Information Technology 32

FAQ . . . . . Frequently Asked Questions 92

GEM . . . . . Gateway to Educational Materials 32

GPL . . . . . GNU General Public License 98

HTML . . . . . Hypertext Markup Language 28

HTTP . . . . . Hypertext Transfer Protocol 53

IBM . . . . . International Business Machines 12

ID . . . . . Identification 26

IE . . . . . Internet Explorer 92

IEC . . . . . International Electronical Commission 24

IEEE . . . . . Institute of Electrical and Electronics Engineers 5

IFS . . . . . Internet File System 96

IIS . . . . . Internet Information Server 90

ILIAS . . . . . Integriertes Lern-, Informations- und Arbeitskooperations-System 81

ILMS . . . . . Integrated Learning Management System 17

IMS . . . . . Instructional Management System 5

ISO . . . . . International Organization for Standardization 24

ISSS . . . . . Information Society Standardization System 27

ITS . . . . . Intelligentes Tutoren-System 12

J2EE . . . . . Java 2 Platform Enterprise Edition 95

JAR . . . . . Java Archive 182

JDBC . . . . . Java Database Connectivity 132

JISC . . . . . Joint Information Systems Committee 109

JPG . . . . . Joint Photographic Experts Group 106

JSP . . . . . Java Server Pages 116

JSTL . . . . . Java Server Pages Standard Tag Library v

JTC . . . . . Joint Technical Committees 29

KI . . . . . Künstliche Intelligenz 12

KMS . . . . . Knowledge Management System 23

LAN . . . . . Local Area Network 11

LCMS . . . . . Learning Content Management System 17

LDAP . . . . . Lightweight Directory Access Protocol 81

LMS . . . . . Learning-Management-System x

LOM . . . . . Learning Objects Metadata 27

LRN . . . . . Learning Resource Interchange 32

LTSA . . . . . Learning Technology System Architecture 29

LTSC . . . . . Learning Technology Standards Committee 24

MIT . . . . . Massachusetts Institute of Technology 31

MOODLE . . . . . Modular Object Oriented Dynamic Learning Environment 104

MS . . . . . Microsoft 32

MS-DOS . . . . . Microsoft-Disk Operating System 12

MVC . . . . . Model-View-Controller 114

NISO . . . . . National Institute of Standards and Technology 25

NUWC . . . . . Naval Undersea Warfare Center 84

OKI . . . . . The Open Knowledge Initiative 31

OSI . . . . . Open Source Initiative 98

OSTP . . . . . White House Office of Science and Technology Policy 35

OUNL . . . . . Open University Niederlanden 31

PAPI . . . . . Public and Privat Information 29

PC . . . . . Personal Computer 12

PDF . . . . . Portable Document Format 83

PHP . . . . . PHP: Hypertext Preprocessor 100

PIF . . . . . Package Interchange File 45

PKZIP . . . . . Phil Katz's ZIP 85

PNG . . . . . Portable Network Graphics 106

PROMETEUS Promoting Multimedia Access to Education and Training in the European Society 30

RBAC . . . . . Role Based Access Control 99

RDF . . . . . Resource Description Framework 28

RTE . . . . . Run-Time Environment 5

RTF . . . . . Rich Text Format 93

SC36 . . . . . Standards for Information Technology for Learning, Education and Training 24

SCO . . . . . Shareable Content Object x

SCORM . . . . . Sharable Content Object Reference Model 5

SDK . . . . . Software Development Kit 90

SMTP . . . . . Simple Mail Transfer Protocol 178

SN . . . . . Sequencing and Navigation 5

SOAP . . . . . Simple Object Access Protocol 78

SQL . . . . . Structured Query Language 90

SS . . . . . Simple Sequencing 59

SSP . . . . . Simple SCORM Player 137

TILE . . . . . The Inclusive Learning Exchange 102

TV . . . . . Television 10

URI . . . . . Uniform Resource Identifier 46

URL . . . . . Uniform Resource Locator 38

W3C . . . . . World Wide Web Consortium 28

WAI . . . . . Web Accessibility Initiative 102

WAN . . . . . Wide Area Network 11

WAR . . . . . Web Application Archive 174

WBL . . . . . Web Based Learning 14

WBT . . . . . Web Based Training 3

WCAG . . . . . Web Content Accessibility Guidelines 102

WEBCT . . . . . Web Course Tools 81

WINDOWS NT . Windows „New Technology“ 92

WINDOWS XP . Windows „eXPerience“ 92

WSL . . . . . Web Supported Learning 14

WWW . . . . . World Wide Web 14

WYSIWYG . . . What You See Is What You Get 103

XML . . . . . Extensible Markup Language 5

XSD . . . . . XML-Schema-Defintion 86

Glossar

Applet bedeutet soviel wie „kleine Applikation“. Verstanden wird darunter meist ein Java-Applet, ein kleines Computerprogramm, das in einem Web-Browser läuft und in der Programmiersprache Java geschrieben ist. [Wik] 100

Application Sharing („Teilen einer Anwendung“) ist eine Ergänzung einer audiovisuellen Konferenz. Hier werden Programme, Daten oder Objekte von zwei oder mehr Beteiligten gleichzeitig genutzt, indem der wechselseitige Zugriff auf einen PC oder die gemeinsame Arbeit auf einem Rechner ermöglicht wird. Das Application Sharing kann passiv (Präsentation, Betrachtung von Objekten) oder aktiv erfolgen, wenn tatsächlich mehrere Beteiligte aktiv an einer Anwendungen arbeiten können. [Wik] 13

Assessment ist im Zusammenhang mit E-Learning ein Beurteilungsverfahren, um die Fähigkeiten und Fertigkeiten (Soft und Hard Skills) sowie den Wissensstand von Lernenden systematisch zu bewerten. [tsy] 100

Asset ist eine elektronische Repräsentation jeglicher Daten, die an einen Web-Client übermittelt werden können (z.B. Text, Bilder, Sound) 41

Authoring bezeichnet in Zusammenhang mit Learning-Management-Systemen Werkzeuge zur Erstellung von Lehrmaterialen. 6

Behaviourismus Beim Behaviourismus wird der Lernene als „Black Box“ betrachtet – inneren Prozessen, die zum Lernen führen wird keine Aufmerksamkeit geschenkt. Man geht davon aus, dass Lernen rein durch Belohnung und Bestrafung gesteuert werden kann. [Blu98] 11

Blackboard oder „Schwarzes Brett“ ist ein Gegenstand, an welchem Informationen ohne weitere Genehmigung angebracht werden können. Häufig sind schwarze Bretter in Form von Pinnwänden oder Tafeln in öffentlichen Ein-

richtungen vorzufinden. [Wik] 6

Black Box Als Black Box bezeichnet man ein Objekt, dessen innerer Aufbau und innere Funktionsweise unbekannt oder als nicht weiter von Bedeutung erachtet wird. Von Interesse ist nur das äußere Verhalten der Black Box. [Wik] 191

Bookmark oder „Lesezeichen“ bezeichnet einen Link, der von einem Computerprogramm zwecks schnellerem Zugriff auf gewisse häufig besuchte Standorte im Internet in einer Lesezeichen-Sammlung verwaltet wird. Meist sind diese Programme sogenannte Browser [Wik] . 82

Broadcast Ein Broadcast in einem Computernetzwerk stellt einen Rundruf dar, wobei Datenpakete von einem Punkt aus gleichzeitig an alle Teilnehmer eines Netzes übertragen werden. [Wik] 13

Browser Ein Programm zur Darstellung der verschiedenen Dokumente aus dem World Wide Web auf einem PC. Die am meisten verbreiteten Browser sind Netscape Navigator und Microsoft Internet Explorer. [tsy] 55

Bulletin ist die Verkleinerungsform von „Bulle“. Als Bulletin werden Bekanntmachungen, Tagesberichte und auch Zeitungen bezeichnet. [Wik] 27

Cache ist ein Zwischenspeicher für Informationen, die voraussichtlich bald erneut abgerufen werden. Daten sind dardurch schneller verfügbar. 55

Chat ist die Bezeichnung für die innerhalb des Internet weit verbreitete Art der direkten Unterhaltung zwischen zwei oder mehreren Personen in Echtzeit, es ist eine Art Computerkonferenz. [Wik] 6

Community Eine Community, der englische Ausdruck für „Gemeinschaft“, ist eine Gruppe von Personen, die gemeinsames Wissen entwickeln, Erfahrungen teilen und dabei eine eigene Identität aufbauen. Communitys profitieren von dem Grundsatz, dass alle Teilnehmer zum Erfolg beitragen, indem sie ihr Wissen einbringen. [Wik] 89

Content ist der informative Inhalt einer Nachricht oder einer Seite, der nicht von Strukturen oder Formaten geprägt ist. Die Bezeichnung ist ein Sammelbegriff für alle medialen Inhalte. [glo] 17

Deployment bezeichnet die Verteilung und Installation von Software auf Zielsysteme einschließlich deren Konfiguration. [Wik] 174

E-Learning (*Electronic Learning*) war ursprünglich ein Sammelbegriff für alle Lern-/Lehrplattformen, die elektronisch durch Informations- und Kommunikationstechnologien unterstützt wurden. Im Zuge des Internet-Hype wurde mit dem Begriff hauptsächlich die Form des netzwerkbasierenden Lernens assoziiert. 3

E-Tutoring (*Electronic Tutoring*) bezeichnet als Form des E-Learnings das Unterrichten, die Unterstützung und die Beurteilung von Studenten mit Hilfe von Online-Technologien. 18

face-to-face Bezeichnet die reale, nicht durch elektronische Medien vermittelte Kommunikation „von Angesicht zu Angesicht“. [uni] 22

Feedback Ein Feedback ist eine Rückmeldung, in der eine Person einer anderen mitteilt, wie sie bestimmte Aussagen, Verhaltensweisen etc. erlebt hat/haben. Im Web bezeichnet man damit ich eine Bewertung bzw. Rückmeldung über eine Website. 7

Framework Wörtlich übersetzt bedeutet Framework „(Programm-)Gerüst, -Rahmen oder -Skelett“. Darin drückt sich aus, dass ein Framework in der Regel – im Gegensatz zu einer reinen Klassenbibliothek – eine Anwendungsarchitektur vorgibt. Beispielsweise legt ein objektorientiertes Framework die Struktur wesentlicher Klassen und Objekte sowie ein Modell des Kontrollflusses in der Applikation fest. In diesem Sinn werden Frameworks im Wesentlichen mit dem Ziel einer Wiederverwendung architektonischer Muster entwickelt und genutzt. [Wik] 7

Intellectual Property bzw. „geistiges Eigentum“ bezeichnet ein Exklusivrecht an immateriellen Gegenständen (Ideen, Werken, Informationen). Dabei werden jedem außer dem Rechteinhaber oder Lizenznehmer ein Verbot bezüglich Verwendung, Nachahmung oder Kopie (Nutzung) der Immaterialgüter auferlegt. [tsy] 28

Internet (*Interconnected Networks*) ist ein weltweites Netzwerk voneinander unabhängiger Netzwerke. Es dient der Kummunikation und dem Austausch von Informationen. Jeder Rechner eines Netzwerkes kann dabei prinzipiell mit jedem anderen Rechner kommunizieren. [Wik] 11

Interoperabilität ist die Fähigkeit unabhängiger, heterogener Systeme möglichst nahtlos zusammen zu arbeiten, um Informationen auf effiziente und verwertbare Art und Weise auszutauschen bzw. dem Benutzer zur Verfügung

zu stellen, ohne dass dazu gesonderte Absprachen zwischen den Systemen notwendig sind. [Wik] 24

Intranet ist ein Rechennetzwerk, das auf den gleichen Techniken wie das Internet basiert, jedoch nur von einer festgelegten Gruppe von Mitgliedern einer Organisation genutzt werden kann. [Wik] 16

Java ist eine objektorientierte, plattformunabhängige Programmiersprache. Java-Programme benötigen zur Ausführung eine spezielle Umgebung, die Java-Laufzeitumgebung oder Java-Plattform genannt wird und als wichtigsten Bestandteil die Java Virtual Machine enthält. Der Vorteil ist, dass nur diese Umgebung an verschiedene Computer und Betriebssysteme angepasst werden muss. Sobald dies geschehen ist, laufen auf der Plattform alle Java-Programme ohne Anpassungsarbeiten. [Wik] 8

JavaScript ist eine von Netscape entwickelte Skriptsprache. Ihr Quellcode wird entweder direkt in HTML-Dokumente geschrieben oder in eine externe Datei ausgelagert. Diese kann in Folge in jedes HTML-Dokument eingebunden werden. JavaScript muss im Gegensatz zu Java nicht kompiliert werden, da der Quellcode zur Laufzeit von einem JavaScript- kompatiblen Web-Browser interpretiert wird. Diese Skriptsprache bietet eine gute Ergänzung und Erweiterung zu HTML, da sie Webseiten mit zusätzlicher Funktionalität versehen kann. [uni] 124

key/value siehe Properties-Datei 177

Klassifikation bezeichnet einen Vorgang oder eine Methode zur Einteilung von Objekten in Klassen oder Kategorien. [Wik] 179

Klassifikator ist eine Verfahrensweise, die eine Zuordnung eines Objekts, dessen Klassenzugehörigkeit unbekannt ist, zu einer Klasse bewerkstelligt. Hierzu muß der Klassifikator zunächst so konstruiert werden, daß er noch nicht klassifizierte Objekte der Klasse zuordnet, zu der sie „am besten passen“. 178

Kognitivismus Bei der kognitivistischen Sichtweise des Lernens spielen die Denk- und Verstehensprozesse der Lernenden eine zentrale Rolle. [Blu98] 12

Konstruktivismus umfasst die Idee, dass Wissen durch eine interne subjektive Konstruktion von Ideen und Konzepten entsteht. [Blu98] 12

Lernpfad Ein Lernpfad besteht aus vorgegebenen Lernschritten, die sicher zum Lernziel und Erfolg führen sollen. Der Begriff wird in erster Linie im Zusam-

menhang mit computergestützten Lernformen verwendet. 19

Makro ist eine Art Textbaustein. Also ein kleines Stück Programmcode, das von einem Interpreter oder Präprozessor durch ein größeres Stück Programmcode ersetzt wird. Somit ist es möglich oft wiederkehrende Programmstrukturen durch Kürzel zu vereinfachen oder litereale Konstanten durch semantische Bezeichnungen zu ersetzen. [Wik] 83

Metadaten Als Metadaten oder Metainformationen bezeichnet man allgemein Daten, die Informationen über andere Daten enthalten. Bei den beschriebenen Daten handelt es sich oft um größere Datensammlungen (Dokumente) wie Bücher, Datenbanken oder Dateien. [Wik] ix

Middleware „Middle“ ist die englische Bezeichnung für „Mittler, Mittelstück“. Middleware ist Software, welche zwei verschiedene Applikationen verbindet – wie z.B. Web Server und Datenbanken. [glo] 134

Monitoring Unter Monitoring versteht man alle Arten der Erfassung von Zuständen, eines Vorgangs oder Prozesses mittels technischer Hilfsmittel oder anderer Beobachtungssysteme. Ein Monitoringsystem ermöglicht Interventionen in die betreffenden Prozesse, sofern sich abzeichnet, dass der Prozess nicht den gewünschten Verlauf nimmt. [Wik] 92

Next Generation Web Das WWW war ursprünglich als Informationssystem gedacht, zugänglich sowohl für Menschen als auch für Maschinen (im weitesten Sinn). Heutzutage ist allerdings der Großteil des Web für Menschen entworfen worden, was die maschinelle Informationsverarbeitung sehr erschwert. Nach Experten soll sich das Web wie es momentan existiert, in ein Semantic Web weiterentwickeln, in dem Wissen in einer Form repräsentiert wird, die sowohl von Menschen als auch von Maschinen „verstanden“ werden kann. Zu diesem Zweck müssen die Daten zunächst in geeigneter Form repräsentiert werden (XML), um dann daraus entsprechende Schlüsse zu ziehen. 11

Open Source Der englische Ausdruck Open Source bzw. „Quelloffenheit“ wird meist auf Computer-Software angewandt und meint, dass es jedem ermöglicht wird, Einblick in den Quelltext eines Programms zu haben. [Wik] 8

Oracle Corporation ist einer der größten Softwarehersteller weltweit mit Hauptsitz in Redwood Shores (Silicon Valley, Kalifornien). Bekanntestes und erfolgreichstes Produkt des Unternehmens ist der Oracle Database Server, welches üblicherweise mit dem Namen Oracle in Verbindung gebracht wird.

[Wik] 95

Parser Als Parser werden Programme bezeichnet, die Dokumente und Quelltexte auf grammatikalische Struktur prüfen. Dies gilt sowohl für normale Dokumente als auch für Quelltexte einer Programmiersprache. [gal] 158

Perl ist eine freie, plattformunabhängige Programmiersprache und wird auch als Skriptsprache bezeichnet. [Wik] 95

Personalisierung bezeichnet im Besonderen das Anpassen eines Dienstes, eines Computerprogrammes oder eines Informationsprodukts an die persönlichen Vorlieben (die Präferenzen), Bedürfnisse, Aversionen oder Fähigkeiten eines Benutzers. [Wik] 16

Plaintext Mit Plaintext („Klartext“) werden Daten bezeichnet, die ausschließlich Text enthalten, der sich dem Leser direkt erschließt ohne dass ein spezielles Programm erforderlich wäre. Plaintext enthält dabei weder binär kodierte Daten noch Metadaten wie etwa Links oder sonstige Auszeichnungen. 152

Plugin („einstöpseln, anschließen“) oder Ergänzungs- oder Zusatzmodul ist eine gängige Bezeichnung für ein Softwareprogramm, das in ein anderes Softwareprodukt ëingeklinkt"wird. [Wik] 81

Präsenzveranstaltung Trainer oder Tutoren und Teilnehmer befinden sich zur gleichen Zeit im gleichen Raum. Einige Fern- und Onlinekurse sind mit Präsenzveranstaltungen kombiniert. [tsy] 18

Properties-Datei Eine Java-Properties-Datei ist eine Plaintext-Datei, die in der Programmiersprache Java als einfacher Konfigurationsmechanismus verwendet wird. Eine Property (zu deutsch „Eigenschaft“) ist in diesem Zusammenhang ein Text („value“), der unter einem bestimmten Namen („key“) registriert ist und auch als „key/value“-Paar bezeichnet wird. Im Struts Framework werden Properties-Dateien zur Internationalisierung von Web-Anwendungen verwendet. [Wik] 120

Pseudocode bezeichnet eine Darstellung eines Quelltextes in einer Sprache, die an eine Programmiersprache erinnert, aber noch leichter lesbar ist als ausformulierter Programmcode. [Wik] 6

Repository Ein Repository ist ein zentraler Ort um Daten zu speichern, zu warten und über ein Netzwerk zu verteilen. 42

Request Ein Request bedeutet, dass ein Client eine Anfrage an einen Server

stellt. Eine solche Anfrage könnte beispielsweise das Anfordern einer bestimmten Website vom Webserver sein. Der Server sendet daraufhin eine Antwort an den Client. Eine solche Antwort nennt man Response. [gal] 65

Response Ein Response folgt immer auf einen Request. Es handelt sich hierbei um eine Antwort auf eine Anfrage. Eine solche Antwort könnte beispielsweise eine Website oder eine Fehlermeldung sein. [gal] 83

Schnittstelle siehe Softwareschnittstelle 179

Schwellwert legt die Grenze der Zuverlässigkeitsinformation fest, anhand der bei einer binären Klassifikation entschieden wird, ob ein Objekt der einen oder der anderen Klasse zugewiesen wird. Üblicherweise ist der Schwellwert bei einer auf Eins normierten Zuverlässigkeitsinformation $\frac{1}{2}$. 179

SCORM (*Sharable Content Object Reference Model*) ist eine von ADL entwickelte Empfehlung zur Standardisierung von Lernobjekten. Das primäre Ziel dabei ist die Bereitstellung eines Referenzmodells für webbasierende Lernmanagementsysteme (LMS) zur Verwendung system- und plattformunabhängiger Lerneinheiten. [Wik] 178

Script oder auch Skriptsprachen sind Programmiersprachen, die vor allem für kleine, überschaubare Programmieraufgaben gedacht sind. Skriptsprachen verzichten oft auf bestimmte Sprachelemente, deren Nutzen erst bei der Bearbeitung größerer Projekte zum Tragen kommen. So wird etwa in vielen Skriptsprachen auf den Deklarationszwang von Variablen verzichtet – vorteilhaft bei kleinen Skripten, bei großen Programmen etwa wegen der fehlenden Überprüfungsmöglichkeit von Tippfehlern in Variablennamen aber von Nachteil. [Wik] 100

Semantik beschäftigt sich als Teilgebiet der Sprachwissenschaft mit dem Sinn und Bedeutung von Sprache. [Wik] 28

Semantic Web ist eine Erweiterung des WWW um maschinenlesbare Daten welche die Semantik der Inhalte formal festlegen. [Wik] 195

Serialisierung bedeutet eine sequentielle Abbildung von Objekten auf eine externe Darstellungsform. Ziel ist das Erreichen von Persistenz für das Objekt. [Wik] 128

Servlet Als Servlet bezeichnet man im Rahmen der J2EE (Java 2 Platform Enterprise Edition) ein Java-Objekt, an das ein Webserver Anfragen seiner Clients

delegiert, um die Antwort an den Client zu erzeugen. Der Inhalt dieser Antwort wird dabei erst im Moment der Anfrage generiert, und ist nicht bereits statisch (etwa in Form einer HTML-Seite für den Webserver verfügbar. [Wik] 116

Shell Eine Shell stellt eine Schnittstelle des Betriebssystems zur Entgegennahme von Befehlen in Form einer Eingabeaufforderung dar. [Wik] 175

Simple SCORM Player ist eine in Kooperation mit [Mit05] erstellte Beispiel-Anwendung einer SCORM-basierenden Lernumgebung. Diese Anwendung dient ausschließlich zur Demonstration der in beider Diplomarbeiten gezeigten Konzepte und ist ausdrücklich nicht als vollwertiges Produkt zu verstehen. 175

Smalltalk ist eine objektorientierte Programmiersprache und zugleich eine vollständige Entwicklungsumgebung, die in den siebziger Jahren am Xerox PARC Forschungszentrum durch Alan Kay, Dan Ingalls, Adele Goldberg und anderen entwickelt wurde. Sie wurde allgemein unter dem Namen Smalltalk-80 freigegeben und hat die Entwicklung vieler späterer Programmiersprachen beeinflusst (z.B. Java). [Lew] 114

Softwareschnittstelle Softwareschnittstellen bieten eine Ebene der Abstraktion, die es erlaubt, auf Funktionen zuzugreifen, ohne ihre Implementation kennen zu müssen. Sie stellen eine formale Deklaration dar, die festlegt, welche Funktionen vorhanden sind (sein müssen). Das hat den Vorteil, dass Module, die die gleiche Schnittstelle besitzen, gegeneinander ausgetauscht werden können. Auch ist es auf dieseWeise möglich, verschiedene Komponenten gleichzeitig zu entwickeln, ohne dass die erste fertig sein muss um die zweite zu übersetzen. Solche Schnittstellen dienen der Modularisierung einer Softwarearchitektur. [Wik] 197

SSP (*Simple SCORM Player*) ist eine in Kooperation mit [Kna05] erstellte Beispiel-Anwendung einer SCORM-basierten Lernumgebung. Diese Anwendung dient ausschließlich zur Demonstration der in beider Magisterarbeiten gezeigten Konzepte und ist ausdrücklich nicht als vollwertiges Produkt zu verstehen. 176

Stand-alone ist die englische Bezeichnung für „allein stehend“. Man Bezeichnet damit eine (unabhängige) Einzelapplikation oder auch einen Einzelplatzrechner. 80

Syllabus beschreibt organisatorische Informationen einer (Lehr-) Veranstaltung und entspricht im Prinzip einem Lehrplan. [Wik] 95

Tag kommt aus dem Englischen und steht für „Etikett“ oder „Auszeichnung“. Neben dem eigentlichen Text enthalten z.B. HTML- oder XML-Dateien spezifische Markierungen, so genannten „Tags“. Tags werden im Quelltext des Dokumentes durch spitze Klammern markiert. Fast alle Markierungen bestehen aus einem einleitenden und einem abschließenden Tag, dazwischen befindet sich der Gültigkeitsbereich. [glo] 130

Tag-Libraries Mit Hilfe von Tag-Libraries ist es möglich, JSP-Seiten zu entwickeln, die nur noch wenig bis gar keinen Java-Code beinhalten. Solche JSPs bieten dann die Schnittstelle zwischen dem Web-Designer, der kein Java versteht, und dem Web-Entwickler, der die dynamischen Teile einer Seite entwickelt. Tag-Libraries können zudem in mehreren JSPs verwendet werden. [Wik] 129

Template der englische Begriff für „Schablone“ bezeichnet eine Vorlage verwendet in Content Management Systemen. [Wik] 6

Thread ist die englische Bezeichnung für „Faden“. Ein gut geschriebener Artikel - z.B. in einer Newsgroup - hat meistens viele Antworten zur Folge. Eine zusammenhängende Kette von Antworten auf einen „Basisbeitrag“ in einer Newsgroup oder einem Diskussionsforum eines Online-Dienstes zu einem ganz bestimmten Thema bezeichnet man als Thread. [glo] 147

Tool ist die englische Bezeichnung für ein Dienstprogramm. 6

Tracer Ein Tracer ist ein Hilfsmittel (Programm, Debug-Funktion,...), mit dem ein Programmablauf oder auch eine Internet-Verbindung beobachtet, protokolliert und zurückverfolgt („getracet“) werden kann. [glo] 199

Tracing siehe Tracer 131

Tracking In Zusammenhang mit Learning-Management-Systemen versteht man unter Tracking eine Überwachung der Sitzungen jedes einzelnen Lerners. Dabei wird beispielsweise registriert, welche Lerninhalte der Lerner sich ansieht, wie er damit arbeitet sowie die Aktivitäten des Lerners im Learning-Management-System insgesamt. 51

Tutor Ein Tutor ist ein Lernbegleiter, dessen Aufgabe es ist, den Lernprozess zu beobachten und im Bedarfsfall helfend einzugreifen. [adl] 6

Tutoring siehe E-Tutoring 8

WebCT (*Web Course Tools*) 31

well-formed Ein XML-Dokument ist well-formed („wohlgeformt“), wenn es den Wohlgeformtheitsregeln gerecht wird. Die Wohlgeformtheit ist Voraussetzung der Typgültigkeit, die zusätzlich die Gültigkeit auf eine vorhandene DTD umfasst. Ein nicht wohlgeformtes Dokument wird nicht als korrektes XML-Dokument betrachtet und von XML-Parsern in jedem Fall zurückgewiesen. [gal] 85

Whiteboard oder „Weißwandtafeln“ sind Tafeln mit einer speziellen Oberfläche aus meist weißem Kunststoff, auf der man mit speziellen Whiteboard-Filzmarkern schreibt. [Wik] 90

Wiki Ein Wiki , auch WikiWiki und WikiWeb genannt, ist eine im World Wide Web verfügbare Seitensammlung, die von den Benutzern nicht nur gelesen, sondern auch online geändert werden kann. Wikis ähneln damit Content Management Systemen. Der Name stammt von „wikiwiki“, dem hawaiianischen Wort für „schnell“. [Wik] 105

Wizard , der englische Begriff für „Zauberer“, ist ein Hilfssystem Für Anwendungen , das den Benutzer Schritt für Schritt durch eine besondere Aufgabe führt. [Lew] 82

Workflow Ein Workflow ist eine Abstraktion der Abfolgen von Tätigkeiten, die vor allem auf den Fluss digitalisierter Dokumente bzw. Objekte gerichtet ist. Menschliche Aktivitäten bzw. Entscheidungen im Rahmen eines Geschäftsprozesses werden dabei weitgehend ausgeklammert bzw. auf Interaktionen mit Anwendungssystemen reduziert. [Wik] 72

WYSIWYG (*What You See Is What You Get*) folgt dem Prinzip, dass die Bildschirm-Darstellung von Programmen auf PCs oder Workstations mit dem späteren Ausdruck möglichst identisch ist. Im Webdesign-Bereich bezeichnet man damit auch HTML-Editoren, bei denen der HTML-Code nicht eingetippt wird, sondern mittels Drag and Drop auf einer graphischen Oberfläche gearbeitet werden kann. Die HTML-Seite wird dabei so dargestellt wie später durch den WYSIWYG Browser. [tsy] 100

XML (*Extensible Markup Language*) ist ein Standard zur Erstellung maschinen- und menschenlesbarer Dokumente in Form einer Baumstruktur. XML defi-

niert dabei die Regeln für den Aufbau solcher Dokumente. Für einen konkreten Anwendungsfall müssen die Details der jeweiligen Dokumente spezifiziert werden. Dies betrifft insbesondere die Festlegung der Strukturelemente und ihre Anordnung innerhalb des Dokumentenbaums. XML ist damit ein Standard zur Definition von beliebigen, in ihrer Grundstruktur jedoch stark verwandten Auszeichnungssprachen. [Wik] 124

ZIP siehe PKZIP 162

Zuverlässigkeitsinformation stellt das Ergebnis einer binären Klassifikation dar. Dabei sollen ein oder mehrere Objekte einer von zwei Klassen zugeordnet werden. Die Zuverlässigkeitsinformation – normiert auf Werte zwischen Null und Eins – beschreibt wie zutreffend die Klassifikation war. Objekte mit Werten zwischen 0 und $\frac{1}{2}$ werden der ersten Klasse zugewiesen, Objekte mit Werten zwischen $\frac{1}{2}$ und 1 der zweiten Klasse. 197

Literaturverzeichnis

[Ack04] ACKERMANN, JAN: *Web Frameworks*. Technischer Bericht, Institut für Wirtschaftsinformatik, Universität Münster, 2004. http://www-wi.uni-muenster.de/pi/lehre/ws0405/seminar/01WebFrameworks.pdf. 115, 134

[adl] *adLexikon* [online, letzter Abruf: 17.09.2005]. http://www.adlexikon.de. 194

[ADL04a] ADLNET.ORG: *SCORM 2004 2nd Edition Overview*. Technischer Bericht, ADLNet.org, Juli 2004. http://www.adlnet.org/downloads/files/67.cfm. 36, 38, 39, 77

[ADL04b] ADLNET.ORG: *SCORM Content Aggregation Model*. Technischer Bericht, ADLNet.org, Juli 2004. http://www.adlnet.org/downloads/files/67.cfm. 41, 42, 43, 44, 45, 46, 77, 150

[ADL04c] ADLNET.ORG: *SCORM Runtime Environment*. Technischer Bericht, ADLNet.org, Juli 2004. http://www.adlnet.org/downloads/files/67.cfm. 50, 54, 56, 77

[ADL04d] ADLNET.ORG: *SCORM Sequencing and Navigation*. Technischer Bericht, ADLNet.org, Juli 2004. http://www.adlnet.org/downloads/files/67.cfm. 59, 60, 64, 70, 72, 74, 77

[ALB04] ANNABELL LORENZ, BIRGIT ZENS und MICHAELA BOCIURKO: *WebCT Vista: Schöne Aussichten*. 2004. Zentraler Informatikdienst der Universität Wien. http://www.univie.ac.at/comment/04-2/042_3.html. 110

[ano05] *e-Learning* [online]. 2005 [letzter Abruf: 16.08.2005]. http://elearning.anova.de. ANOVA Multimedia Studios GmbH Rostock. 7, 25, 40, 77

[Bau02] BAUMGÄRTNER, THOMAS: *Lehren und Lernen mit Neuen Medien (Multimedia) in der universitären Ausbildung - Entwicklung und Evaluation eines multimedialen Tauch-Lern-Systems*. Diplomarbeit, Fakultät für Geistes- und Sozialwissenschaften,Universität Karlsruhe, 2002. http://www.ubka.uni-karlsruhe.de/vvv/2002/geist-soz/1/1.pdf. 7

[Bee03] BEEGER, ROBERT F.: *Herstellung von Konsistenz und Validität in Web-Anwendungen*. Diplomarbeit, Fakultät fur Informatik und Automatisierung, Technischen Universität Ilmenau, 2003. http://swt5.informatik.uni-hamburg.de/publications/files/Dipl/rb_dipl.pdf. 123, 134

[Blu98] BLUMSTENGEL, ASTRID: *Entwicklung hypermedialer Lernsysteme*. Vortragsfolien, Juli 1998. Decision Support and Operations Research Lab, Universität Paderborn. ttp://dsor.uni-paderborn.de/de/forschung/publikationen/blumstengel-diss/main_index_titel.html. 18, 33, 186, 189

[Che04] CHEVALIER, PIERRE-ANDR´E: *Eine kurze Einleitung in das Lernen*. 2004. Hochschule für Technik und Informatik, Berner Fachhochschule. http://www.hti.bfh.ch/fileadmin/img/HTI/div-marti/didaktik/lernen.pdf. 7

[dASF01] APACHE SOFTWARE FOUNDATION, DAS STRUTS-FRAMEWORK DER:

Sebastian Stein, 2001. Studienrarbeit. http://www.sebastian-stein.de/media-center/content/Struts-Framework%20-%20Endfassung.pdf. 134

[Dec04] DECKER, DENISE: *SCORM Sequencing und Navigation - Ein Vergleich zwischen SCORM Version 1.2 und Version 1.3*. Seminararbeit, Universität Mannheim, Jänner 2004. http://www.wifo.uni-mannheim.de/~mazloumi/seminar/decker-2004-scorm-sn-vgl.pdf. 77

[Edu] EDUTOOLS. *Course Management Systems* [online, letzter Abruf: 22.08.2005]. http://www.edutools.info. 110

[est] *eLearning Plattformen in Hamburg* [online, letzter Abruf: 18.08.2005]. http://www.estudent-hamburg.de/. 92, 110

[ete] *Portal e-teaching.org* [online, letzter Abruf: 19.08.2005]. http://www.e-teaching.org/. 110

[exe05] *the eXe project* [online]. August 2005 [letzter Abruf: 17.08.2005]. http://exe.cfdl.auckland.ac.nz/. The University of Auckland. 89

[Exn05] EXNER, TORSTEN: *e-Learning Plattformen und Standards*. Diplomarbeit, Fachhochschule Worms, 2005. http://www.fh-worms.de/bib-fh/edocs/diplinf/2005/exner-torsten-diplomarbeit.pdf. 33, 77, 110

[Fes03] FEST, SUSANNE: *Technische Szenarien für den e-Learning Einsatz im Ergänzungslehrgang der Fachhochschul-Studiengänge Oberösterreich*. Diplomarbeit, Medientechnik und -design und Digitale Medien, FH Hagenberg, 2003. http://cblinux.fh-hagenberg.at/links/Fest-2003/Diplomarbeit_mtd99015.pdf. 110

[For04] FORD, NEAL: *Art of Java Web Development*. Manning Publications Co., 2004. 134

[Fro] FROTSCHER, THILO. *Das nächste Spiel ist immer das schwerste - Der Ball ist rund, Teil 2: Betrachtung der Präsentationslogik beim Erstellen eines Tippspiels* [online, letzter Abruf: 09.09.2005]. http://www.sqlkon.net/itr/online_artikel/psecom,id,581,nodeid,11.html. 134

[gal] *Galileo Computing* [online, letzter Abruf: 17.09.2005]. http://www.galileocomputing.de/glossar. 191, 192, 195

[GEGB04] GERALD EIGENSTUHLER, FELIX MÖDRITSCHER und VICTOR MANUEL GARCÍA-BARRIOS: *Evaluierung von Open Source E-Learning-Plattformen, Untersuchungen im Rahmen des Forschungsprojektes AdeLE (Adaptive e-Learning with Eye-tracking)*. Technischer Bericht, Institut für Informationssysteme und Computer Medien, Technische Universität Graz, 2004. http://www2.iicm.edu/cguetl/research/opensource_elearning_systems. 110

[Gel05] GELLWEILER, JAN: *Die ADL SCORM Spezifikation und ihre Verbreitung*. Vortragsfolien, Juli 2005. Fern-Universität Hagen, Lehrgebiet Kommunikationssysteme. http://ks.fernuni-hagen.de/aktivitaeten/praesentationen/pubs/SCORM_Spezifikation_und_Verbreitung.pdf. 32, 33, 77, 84, 86, 110

[Ger04] GERICKE, ANNE-KATHRIN: *Untersuchung und Bewertung von SCORM 2004 sowie prototypische Realisierung von Online-Kursmaterial nach dieser Spezifikation* . Diplomarbeit, Hochschule für Technik und Wirtschaft Dresden (FH), Fachbereich Informatik/Mathematik, 2004. http://www.merino.de/_download/diplomarbeiten/gericke-diplomarbeit.pdf. 7

[glo] *ARCHmatic-Glossar und -Lexikon* [online, letzter Abruf: 12.09.2005]. http://www.glossar.de. 187, 190, 194

[Glü04] GLÜCKSELIG, STEFFEN: *Standards e-Learning Umfeld: SCORM*. 2004. Institut für Informatik, Universität Würzburg. http://www.gungfu.

de/studium/e-learning_scorm/SCORM_2004_Ausarbeitung.pdf. 77

[HA04] HABIB AZIMI, DIRK DÜSTERHÖFT, CHRISTIAN KNUTH ET AL.: *Projektbericht eLearning, Dokumentation zum studienbegleitenden Projekt "eLearning"*. Technischer Bericht, Multimedia Service Point und ZBRV der Hochschule Bremerhaven, 2004. http://www.hs-bremerhaven.de/Binaries/Binary3920/Projektbericht_eLearning_040429.pdf. 7

[Han02] HANIMANN, SABINE: *Einbindung von e-Learning in den universitären Lehrbetrieb*. Diplomarbeit, Institut für Betriebswirtschaftliche Forschung, Lehrstuhl Marketing, Universität Zürich, 2002. http://cblinux.fh-hagenberg.at/links/Fest-2003/Diplomarbeit_mtd99015.pdf. 33

[HB03] HEIKO BUCHHOLZ, MARKUS E. LEYPOLD ANDTORSTEN SCHILLING: *Lehr- und Lernmanagementsysteme im Vergleich, ein Überblick zur Entscheidungshilfe für das Rechenzentrum der Universität Rostock*. Technischer Bericht, Lehrstuhl Rechnerarchitektur, Fachbereich Informatik, Universität Rostock, 2003. http://www.campussource.de/aktuelles/docs/lms-im-vergleich-UniRostock.pdf. 110

[HG03] HOLZMANN, CLEMENS und THOMAS GÖBL: *MobiLearn - Der ADL SCORM-Standard*. Technischer Bericht Version 1,1, Universität Linz, Universität Wien, Technische Universität Wien, Universität Klagenfurt, 2003. http://www.mobilearn.at/ergebnisse/workp2/standards/ADL%20SCORM%20v1.1.pdf. 77

[HK03] HETTRICH, ALEXANDER und NATASCHA KOROLEVA: *Marktstudie, Learning Management Systeme (LMS) und Learning Content Management Systeme (LCMS), Fokus deutscher Markt*. Technischer Bericht, Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO, Stuttgart, 2003. http://www.iltec.de/downloads/IAOLMSLCMSStudie.pdf. 110

[HW04] HOHENSTEIN, ANDREAS und KARL WILBERS: *Handbuch E-Learning, Expertenwissen aus Wissenschaft und Praxis*. Technischer Bericht, Verlag "Deutscher Wirtschaftsdienst", Wolters Kluwer Deutschland GmbH, Köln, Juli 2004. http://www.elearning-reviews.org/topics/technology/strategic-issues/2004-kiedrowski-open-source-software.pdf. 110

[Häf03] HÄFELE, HARTMUT: *E-Learning Standards, betrachtet aus der didaktischen Perspektive*. Technischer Bericht, Arge Virtual-Learning, 2003. http://www.qualifizierung.com/download/files/e-learning-standards.pdf. 7, 33

[IEE02] IEEE: *Draft Standard for Learning Object Metadata*. Technischer Bericht, Juli 2002. http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf. 77

[ili05] *Welcome to ILIAS Doc - the online documentation of the ILIAS learning management system* [online]. Februar 2005 [letzter Abruf: 17.08.2005]. http://www.homer.ilias.uni-koeln.de/iliasdoc/doc/html/. 99, 110

[imc04] IMC: *CLIX 4.5, Corporate Learning and Information eXchange*. 2004. imc, Advanced Learning Solutions. http://www.clickandlearn.at/fileadmin/user_upload/Ressourcen/Pdf/CLIX-Produktinformation.pdf. 110

[inf] *ELAN eLearning Infothek* [online, letzter Abruf: 27.08.2005]. http://www.l3s.de/elan/kb3/index.php. Forschungszentrum L3S. 7

[Jen04] JENDRYSCHIK, MICHAEL. *Seminar Computer und Gesellschaft* [online]. Juli 2004 [letzter Abruf: 27.08.2005]. http://jendryschik.de/michael/inf/. 7

[Kai01] KAISER, RONALD: *Analyse und Anwendung von Standards für e-Learning-Umgebungen unter besonderer Berücksichtigung des SCORM-Modells*. Diplomarbeit, Hochschule für Technik und Wirtschaft Dresden (FH), 2001. http://cblinux.fhs-hagenberg.ac.at/links/Diplomarbeit08pdf.pdf. 14, 16, 33

[Kna05] KNALL, THOMAS: *Automatische Adaptierung von SCORM-basierenden Lerninhalten*. Diplomarbeit, Technische Universität Graz, Institut für Informationssysteme und Computer Medien (IICM), A-8010 Graz, September 2005. http://www.iicm.edu/thesis/tknall.pdf. H, 7, 153, 193

[Koc02] KOCH, LEIF M.: *Wissensportalsoftware als Learning Management System, Fachliche Analyse, Entwurf, prototypische Realisierung*. Diplomarbeit, Technische Universität Hamburg-Harburg, 2002. http://www.sts.tu-harburg.de/people/r.f.moeller/lectures/anatomie-i-und-k-system/Koch-02.pdf. 33, 110

[Krü04] KRÜGER, GUIDO: *Handbuch der Java-Programmierung*. Addison-Wesley, 4 Auflage, 2004. http://www.javabuch.de. 134

[Kun03] KUNZE, TORSTEN: *Untersuchung zur Realisierbarkeit einer technologieneutralen Mapping-Schicht fur den Datenaustausch am Beispiel einer Anwendung zum medizinischen Formulardruck als integrativer Bestandteil eines Electronic Health Record (EHR)*. Diplomarbeit, Fakultät fur Informatik und Automatisierung, Technischen Universität Ilmenau, 2003. http://cybop.berlios.de/papers/2003_flexible_communication_layer_diploma/Diplomarbeit.pdf. 134

[Küp02] KÜPPERS, FRIEDHELM: *Programmiertes versus persönliches Online-Tutoring*. Technischer Bericht, Euro-Schulen Hannover GmbH, 2002. http://www.ag-fernstudium.de/Tagung/kuepp1.pdf. 33

[Lea04] LEARNING TECHNOLOGY STANDARDS COMMITTEE OF THE IEEE:

RELOAD Editor introductory Manual, IEEE 1484.12.1-2002 Auflage, Juli 2004. http://www.reload.ac.uk/ex/editor_v1_3_manual.pdf. 110

[Lea05] LEARNING TECHNOLOGY STANDARDS COMMITTEE OF THE IEEE: *Blackboard Benutzerhandbuch*, Application Pack 3 (Release 6.3) Auflage, Juni 2005. http://www.blackboard.com/docs/r6/6_3/de_DE/student/bbas_r6_3_user/. 110

[Lew] LEWANDOWSKI, WOLFGANG. *Das Computerlexikon* [online, letzter Abruf: 13.09.2005]. http://www.dascomputerlexikon.de. 193, 195

[LS04a] LIPPOTH, KARL ULRICH und MANFRED SCHWERES. *E-Learning braucht Kontinuität. Mehr nicht?* [online]. Februar 2004 [letzter Abruf: 28.08.2005]. http://www.heise.de/tp/r4/html/result.xhtml?url=/tp/r4/artikel/16/16782/1.html&words=learning. 7

[LS04b] LUX, MATHIAS und GEORG SCHARRER: *Evaluierung der E-Learning-Plattform ATutor*. Technischer Bericht, Institut für Informationssysteme und Computer Medien, Technische Universität Graz, 2004. Multimediale Informationssysteme Übungen WS 2003/04. http://www2.iicm.edu/cguetl/research/opensource_elearning_systems/Ausarbeitungen/Atutor/Atutor.pdf. 110

[MFS03] MARTIN FISCHER, PHILIPP GROLLMANN, BIBHUTI ROY und NIKOLAUS STEFFEN: *E-Learning in der Berufsbildungspraxis: Stand, Probleme, Perspektiven*. Technischer Bericht, Abteilung: Informationstechnik und Kompetenz, Universität Bremen, 2003. http://www.itb.uni-bremen.de/downloads/fb_06_03.pdf. 33

[MG04] MATTHIAS GRUMER, MIOCHAEL JAINDL, ALEX JANEK: *Evaluierung der E-Learning-Plattform Moodle*. Technischer Bericht, Institut für Informationssysteme und Computer Medien, Technische Universität Graz, 2004. Multimediale Informationssysteme Übungen WS 2003/04. http://www2.iicm.edu/cguetl/research/opensource_elearning_systems/Ausarbeitungen/Moodle/Moodle.pdf. 110

[MHH02] MAIER-HÄFELE, KORNELIA und HARTMUT HÄFELE: *Learning-, Content- und Learning-Content-Management-Systeme: Gemeinsamkeiten und Unterschiede*. Technischer Bericht, Arge Virtual-Learning, 2002. http://www.qualifizierung.com/download/files/LMS-CMS-LCMS.pdf. 33, 110

[min] *WebCT Vista - University of Minnesota* [online, letzter Abruf: 18.08.2005]. http://webct.umn.edu/. 95, 110

[Mit05] MITTER, MICHAEL: *Implementierung eines SCORM-basierenden Zugangs zu Web-basierendem Unterricht*. Diplomarbeit, Technische Universität Graz, Institut für Informationssysteme und Computer Medien (IICM), A-8010 Graz, September 2005. http://www.iicm.edu/thesis/mmitter.pdf. 193

[Mon04] MONTANDON, CORINNE: *Standardisierung im e-Learning, Eine empirische Untersuchung an Schweizer Hochschulen*. Technischer Bericht, Institut für Wirtschaftsinformatik der Universität Bern, 2004. http://www.ie.iwi.unibe.ch/publikationen/berichte/resource/No161.pdf. 33

[moo] *Labor Online Learning - Technische Fachhochschule Berlin* [online, letzter Abruf: 18.08.2005]. http://lms.tfh-berlin.de/moodle/. 105, 110

[Nie03] NIESLONY, ARTHUR: *Intelligente Tutorielle Systeme*. Technischer Bericht, Lehrstuhl fur Praktische Informatik IV, Fakultät fur Mathematik und Informatik, Universität Mannheim, 2003. 20, 33

[NP03] NORTON, MARK und ANGELO PANAR: *IMS Simple Sequencing Information and Behavior Model, Version 1.0 Final Specification*. Technischer Bericht, IMS Global Learning Consortium, Burlington, USA, März 2003. http://www.imsglobal.org/simplesequencing/ssv1p0/imsss_infov1p0.html. 77

[Per05] PERENS, BRUCE. *The Open Source Definition* [online]. 2005 [letzter Abruf: 15.08.2005]. http://www.opensource.org/. Open Source Initiative (OSI). 110

[Pet04] PETERS, MEIKEL: *Begriffsauffassungen und Entwicklungsgeschichte des E-Learnings*. Seminararbeit, Institut für Wirtschaftsinformatik, Universität Hannover, 2004. http://www.iwi.uni-hannover.de/historie/peters/E-Learning.pdf. 33

[Pri03] PRITZKAU, ALBERT: *Aktualisierung und Anpassung von E-Learning Content auf der Basis von XML*. Technischer Bericht, Didaktik der Informatik und E-Learning, Universität Siegen, 2003. http://www.die.informatik.uni-siegen.de/lehre/ws_2002_03/proseminar_e-learning/Aktualisierung_und_Anpassung_von_E_Learning_Content_auf_der_Basis_von_XML__Ausarbeitung.pdf. 7

[Rau01] RAUTENSTRAUCH, CHRISTINA: *Tele-Tutoring, Zur Didaktik des kommunikativen Handelns im virtuellen Lernraum*. Technischer Bericht, Institute of educational theory and computer Science, University of Bielefeld, 2001. http://www.uni-hildesheim.de/~chlehn/isko2001/texte/rautenstrauch.pdf. 33

[Rau03] RAUCH, HANS. *Unterschiedliche Arten von E-Learning* [online]. Jänner 2003 [letzter Abruf: 21.07.2005]. http://medien.bildung.hessen.de/online-dienste/lbo/ezine/1016542473. 33

[Rie02] RIEKHOF, HANS-CHRISTIAN: *eLearning und Wissensmanagement in deutschen Großunternehmen*. 2002. Bundesinstitut für Berufsbildung, Bonn. http://www.bibb.de/de/limpact13015.htm. 33

[RKMH05] ROBERT KRISTÖFL, PETER BAUMGARTNER, HARTMUT HÄFELE und KORNELIA MAIER-HÄFELE: *Evaluation von Lernplattformen: Verfahren, Ergebnisse und Empfehlungen (Version 1.3)*. Technischer Bericht, Im Auftrag des Bundesministeriums für Bildung, Wissenschaft und Kultur (BMBWK), Arge virtual-learning, 2005.

http://www.bildung.at/statisch/bmbwk/lernplattformen_evaluation_und_ergebnisse_1_bis_3.pdf. 110

[Sch04a] SCHEUERER, PATRICK: *CTC Analytics Support Portal - Ein Content Management System mit Jakarta Struts und Object/Relational Bridge*. Diplomarbeit, CTC Analytics AG, Fachhochschule beider Basel, 2004. http://www.fhbb.ch/tools/projekte/pdf/0105_877_DA13_03_Arbeit.pdf. 134

[Sch04b] SCHMIDT, ANSGAR: *Sharable Content Object Refernce Model (SCORM)*. Technischer Bericht, M.I.T newmedia GMBH, 2004. http://www.mit.de/downloads/scormartikel_ash_safe.pdf. 77

[Sch04c] SCHNEIDER, ARNE: *IMS Learning Design als Grundlage für die Gestaltung von e-Learning-Systemen*. Diplomarbeit, Fachbereich Wirtschaftswissenschaften, Johann Wolfgang Goethe-Universität, Frankfurt am Main, 2004. http://dspace.learningnetworks.org/retrieve/460/diplomarbeit_ims_ld.pdf. 77

[Sch05] SCHULMEISTER, ROLF: *Zur Didaktik des Einsatzes von Lernplattformen*. Technischer Bericht, Interdisziplinäres Zentrum für Hochschuldidaktik, Universität Hamburg, 2005. http://www.izhd.uni-hamburg.de/pdfs/Lernplattformen.pdf. 110

[Slo02] SLOSSER, STEVE: *SCORM High-level Briefing*. Vortragsfolien, Dezember 2002. http://www.jointadlcolab.org/ADL%20and%20SCORM.ppt. 77

[sof] *Encyclopedia of free software* [online, letzter Abruf: 18.08.2005]. http://www.softpedia.com/. 102, 110

[Ste02] STEINBACH, TORSTEN: *Database Connection Pooling in J2EE*, Band 2, Seiten 48–52. Dpunkt Verlag, 2002. http://mordor.prakinf.tu-ilmenau.de/papers/dbspektrum/dbs-02-48.pdf. 132, 134

[str] *The Apache Software Foundation, Struts* [online, letzter Abruf: 10.09.2005]. http://struts.apache.org. 134

[TES04] THOMAS EDLINGER, DAVID FEINER und MICHAEL SCHARNREITNER: *Evaluierung der E-Learning-Plattform ILIAS*. Technischer Bericht, Institut für Informationssysteme und Computer Medien, Technische Universität Graz, 2004. Multimediale Informationssysteme Übungen WS 2003/04. http://www2.iicm.edu/cguetl/research/opensource_elearning_systems/Ausarbeitungen/Ilias/Illias.pdf. 110

[TNHW03] TED N. HUSTED, CEDRIC DUMOULIN, GEORGE FRANCISCUS und DAVID WINTERFELDT: *Struts in Action, Building web applications with the leading Java framework*. Manning Publications Co., 2003. 134

[Tor04] TORENS, CHRISTOPH: *SCORM und das semantische Web*. Seminararbeit, Jänner 2004. Institut für Betriebssysteme und Rechnerverbund, TU Braunsschweig. http://www.ibr.cs.tu-bs.de/lehre/ws0304/svs/work/scormweb_paper_final.pdf. 77

[tsy] *Portal Global Learning* [online, letzter Abruf: 07.09.2005]. http://www.global-learning.de. T-System Multimedia Solutions GmbH. 186, 187, 188, 191, 195

[TZ03] TERGAN, SIGMAR-OLAF und PETER ZENTEL: *Lernplattformen und die Zukunft des E-Learning*. Technischer Bericht, IWM - Institut für Wissensmedien, Tübingen, 2003. http://www.iwm-kmrc.de/kevih/workshops/plattformmat/Tergan.pdf. 166

[Ull04] ULLENBOOM, CHRISTIAN: *Java ist auch eine Insel - Programmieren für die Java 2-Plattform in der Version 5 (Tiger-Release)*. Galileo Computing, 4 Auflage, 2004. http://www.galileocomputing.de/openbook/javainsel4/. 134

[uni] *Virtuelles eLearningCenter der Universität Wien* [online, letzter Abruf:

22.08.2005]. http://elearningcenter.univie.ac.at. 7, 110, 188, 189

[Uni05] UNIVERSITY, CARNEGIE MELLON: *SCORM, Best Practices Guide for Content Developers 1st Edition*. Technischer Bericht, Learning Systems Architecture Lab Carnegie Mellon University, Pittsburgh, April 2005. http://www.lsal.cmu.edu/lsal/expertise/projects/developersguide/developersguide/guide-v1p1-20050405.pdf. 77

[Wik] WIKIPEDIA. *deutsche Ausgabe der freien Enzyklopädie Wikipedia* [online, letzter Abruf: 16.08.2005]. http://de.wikipedia.org/. 33, 77, 110, 134, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196

[Zum02] ZUMBACH, JÖRG: *eTutoring - Aufgaben und Anforderungen an ein neues Betätigungsfeld*. 2002. Psychologischen Institut der Universität Heidelberg. http://zumbach.psi.uni-heidelberg.de/pubs/zumbachjournal05.pdf. 33

Index

A

Ablaufsteuerung ... *siehe auch* Sequencing und Navigation
Action ... 122
ActionForm ... 119
ActionMapping ... 122
ActionServlet ... 121
Activity ... 60, 148
Activity Tree ... 60
Adapterklasse ... 179
Adaptoring ... *siehe* Personalisierung
ADL Initiative ... 35
Aktion ... *siehe* Activity, 153
Aktionsbaum ... *siehe* Activity Tree
Algorithmus
  alternativer ... 179
  Vector Space Search ... 179
Anwenderschnittstelle ... 77
Apache
  Ant ... 181
  Commons ... 173
  HTTP Server ... 171
  Lucene ... 172
  Struts ... 172
    Menu Tag Library ... 172
  Tomcat ... 172, 175
    Webanwendungs-Manager **176**, 178, 181
API ... *siehe* Application Programming Interface, 54
  Adapter ... 54
  Funktionen ... 54
  Instanz ... 55
Application Programming Interface 51, 54
Assessment ... 78
Asset ... 41
Attempt ... 61
ATutor ... 102
Authoring ... 153
Automatisierung ... 33
Autorensystem ... 16

B

Baumstruktur ... 44
Behaviourismus ... 11
Beitrag
  neuer ... 179
Betreff ... 178
Blackboard ... 89
  Building Blocks ... 90
Business Logic Beans ... 118
Bytecode-Dateien ... 182

C

CBT ... *siehe* Computer Based Training
Chat ... 23
Classifier4J ... 172, 179
CLIX ... 92
Cluster ... 60
CMS .. *siehe* Content Management System
Computer Based Instruction ... 11
Computer Based Training ... 11, 14
Connection-Pooling ... 133, 160
  DataSources ... 133
Container ... 145, 152
Content Aggregation Model
  Content Model ... **41**
  Content-Packaging ... **43**
  Meta-Data ... **47**
Content Management System ... 16, 17
Content Organization ... 43
Content Package ... 45
Content Packaging ... 153
Content-Packaging ... 43
  Content Hierarchy ... 44
  Content Specific Meta-Data ... 44
  Sequencing and Navigation ... 45
Controller ... 121
CopperCore ... 110

D

Daten-
Austausch ... 30
Datenbank ... 132
MS Access ... 160, 162
MySQL ... 160
Datenmodell ... *siehe* Model
Datentransfer ... 54
Diskussionsforum
Activity ... 148
Thread ... 147
Document Object Model ... 53
DOM ... *siehe* Document Object Model
DTO ... *siehe* Dynamic Transfer Object
DynaActionForm ... 125
Dynamic Transfer Object ... 127
DynaValidationForm ... 127

E

E-Learning ... 10
asynchrones ... 13
Content ... 34
Entwicklung ... 11
Gremien ... 24
Initiativen ... 24
Standards ... 24
synchrones ... 13
Typen ... 13
E-Mail ... 13
Adresse ... 178
E-Tutoring ... 18, 19
EL ... *siehe* Expression Language
Expression Language ... 130

F

face-to-face ... 22, 33

G

GNU General Public License ... 98
GPL ... *siehe* GNU General Public License

H

HTTP ... *siehe* Hypertext Transfer Protocol
Hypertext Transfer Protocol ... 53

I

ILIAS ... 99
Intelligentes Tutoren-System ... 12
Internationalisierung ... 120
Interoperabilität ... 24, 30
Item ... *siehe* Lektion, 152
ITS ... *siehe* Intelligentes Tutoren-System

J

J2SE ... 172
Java-Bean ... 117
JavaMail API ... 173
JDBC ... 132
Connection ... 132
Treiber ... 134
JDOM ... 172
JSTL ... 129

K

Künstliche Intelligenz ... 12
KI ... *siehe* Künstliche Intelligenz
Klassifikator ... 179
KMS ... *siehe* Knowledge Management System
Knowledge Management System ... 23
Kognitivismus ... 12
Kubuntu Linux ... 171
Kurs ... 152
Administration ... 29
Austausch ... 29
Kombination ... 29
Kurs, importierter ... 177
Kurs-
Import ... 178
Validierung ... 178

L

Learning Management System ... 11, **15**
Learning Objects Metadata ... 27, 49
Learning Technology Standards Committee ... 24
Learning-Management-System ... 80
Überblick ... 107
Anforderungen ... 81
Evaluierung ... 83
kommerzielle ... 89
Blackboard ... 89
CLIX ... 92
WebCT ... 94
Open Source ... 97
ATutor ... 102
ILIAS ... 99
Moodle ... 104
Personalisierung ... 82
Studie ... 80
Templates ... 93
Lektion ... 146, 152

Lern-
Automatisierung ... 30
Kontrolle ... 23
Modul ... 29
Objekt
Einbettung ... 31
Pfad ... 19
Plattform ... **16**
Portal ... **15**
Ressource
austauschen ... 43
Ressourcen ... 30
Software ... 23
Steuerung ... 19
Typ
handlungsorientierter ... 23
Umgebung ... **15**, 30
Lernen
aktives ... 14
elektronisch unterstütztes ... 10
elektronisches ... *siehe* E-Learning
netzwerkbasierend ... 11
selbständiges ... 14
selbstgesteuertes ... 22
Lernpfad ... 153
Lernumgebung ... *siehe* Learning-Management-System
Lernziele ... *siehe* Learning Objectives
Limit ... 179
LMS ... *siehe* Learning Mamagement System, *siehe auch* Learning-Management-System
Log4j ... 131, 173
Appender ... 131
Level ... 131
LOM ... *siehe* Learning Objects Metadata, *siehe* Learning Objects Metadata
LTSC *siehe* Learning Technology Standards Committee

M

Manifest ... 30, 46
Manifest Maker ... 109
MessageBoard ... *siehe* Diskussionsforum
Meta-Data ... 47
Metadaten ... 16, *siehe* Meta-Data, 51
DCMES ... 28
DCMI ... *siehe* Standard
Standard ... 28
Mindest-
Ähnlichkeit ... 179
MIT ... 31
Model-View-Controller ... *siehe* MVC
Modul
Anmeldung ... 139
Benutzerverwaltung ... 140
Diskussionsforum ... 147
Experten- ... 20
Kommunikations- ... 20
Kurslektionen ... 145
Container ... 145
Item ... 145
Kursverwaltung ... 142
Studenten- ... 20
Unterrichts- ... 20
Moodle ... 104
MVC
Model 1 Architektur ... 116
Model 2 Architektur ... 116
MVC-Paradigma ... 114
MySQL
Connector/J ... 172
Database Server ... 172
Zugangsdaten ... 176
Skript ... 176

N

Next Generation Web ... 11, 15

O

Online-Meeting ... 23
Open Source ... 97
GPL ... 98
OSI ... 98
Open Source Initiative ... 98
OSI ... *siehe* Open Source Initiative

P

Package Interchange File ... 45
Passwort ... 174, 178
Personalisierung ... 16, 29
Personen
Elliott Masie ... 10
Martin Dougiamas ... 104
Matt Raible ... 128
Skinner, B.F. ... 11
Thomas Knall ... 155
Pfad
absoluter ... 177
relativer ... 178
Pfad-Angaben ... 176
PIF ... *siehe* Package Interchange File

Plaintext-
Dateien ... 177
Präsentationsschicht ... *siehe* View
Prepared Statements ... 134, 160
Properties-Datei ... 177, 179

Q

Quelltext ... 182

R

Reload Editor ... 109
Reload Player ... 109
Repository ... 47
RTE ... *siehe* Run-Time Environment
Run-Time Environment ... 42, **51**
API ... 54
Data Model ... 57
Launch Mechanism ... 52
Asset ... 52
SCO ... 53

S

Sample RTE ... 172
SC36 ... 24
Schablonen ... *siehe* Templates
Schlüsselworte
`$CATALINA_HOME` ... 175
`MessageSimilarity` ... 179
`MessageSimilarityAdapter` ... 179
`MessageSimilarityDummy` ... 179
`SCORMplayer` ... 176
Schnittstelle ... 179
Schwellwert ... 179
SCO ... *siehe* Shareable Content Object
Tracking ... 57
Zustand ... 54, 55
SCORM ... 37
ADL Initiative ... 35
Adopters ... 89
Architektur ... 40
Content Aggregation Model ... 47
Einschränkungen ... 77
Geschichte ... 39
Navigation Model ... 74
Run-Time Navigation Data Model ... 74
Overall Sequencing Process ... 72
Run-Time Environment ... 51
Launch Mechanism ... 52
Sequencing ... 163
Sequencing and Navigation ... 59
Sequencing Control Modes ... 163
Tracking Model ... 70
Zertifizierung ... 84
SCORM 1.2 ... 84
SCORM 2004 ... 86
SCORM Visualizer ... 110
Sequencing
Activity ... 60
Activity Tree ... 60
Cluster ... 60
Learning Objectives ... 62
Overall Sequencing Process ... 72
Sequencing Definition Model ... 62
Constrain Choice Controls ... 66
Delivery Controls ... 70
Limit Conditions ... 67
Randomization Controls ... 69
Rollup Consideration Controls ... 69
Rollup Rules ... 68
Sequencing Control Modes ... 63
Sequencing Rules ... 66
Sequencing Session/Loop ... 72
Sequencing and Navigation ... **45**, 59
Attempts ... 61
Version 1.2 ... 75
Sequencing Control Modes ... 63
Shareable Content Object ... 42
Shell ... 175
Simple Object Access Protocol ... 78
Simple SCORM Player ... 137
Datenbank-Struktur ... 160
Deployment ... 174
Einrichtung ... 174
Entpacken ... 175
Implementierung ... 156
Individuelles Tutoring ... 151
Konzept ... 152
Problemstellung ... 151
Workflow ... 153
Installation ... 175
Konfiguration ... 176
Konfigurationsdatei ... 176, 177
Module ... 139
Neukompilierung ... 181
Paket-Hierarchie ... 158
Requirements ... 137
SCORM
Einschränkungen ... 164
Importvorgang ... 162
Integration ... 162

Sequencing 163
Voraussetzungen 175
SMTP-Server 178
SOAP . *siehe* Simple Object Access Protocol
Software, verwendete 171
SSP ..... *siehe* Simple SCORM Player
Standalone Applikationen 109
CopperCore 110
Manifest Maker 109
Reload Editor 109
Reload Player 109
SCORM Visualizer 110
THESIS 109
Standard
ADL **26**
AICC **25**
ARIADNE **27**
CanCore 32
CEN/ISSS **27**
DCMI **28**
EML 28, **31**
GEM 32
IMS **30**
JTC SC36 ..... **29**
LRN 32
LTSC **28**
CMI 29
LOM 29
LTSA 29
PAPI 29
OKI **31**
PROMETEUS **30**
SCORM 26, **35**
Standardisierungsgremien 24
Stoppwort
Entfernung 179
Listen 179, **181**
Struts ..... *siehe* Struts Framework
Struts Framework 114, 156
Connection-Pooling 133
Controller 121
Action 122
ActionMapping 122
ActionServlet 121
Datenbank-Anbindung 132
DynaActionForm 125
Dynamic Transfer Object 127
DynaValidationForm 127
Expression Language 130
JDBC 132
JSTL 129
Log4j 131
Model 117
ActionForm-Bean 119
Business Logic Beans 118
System State Beans 118
Modifikationen und Erweiterungen ... 124
Prepared Statements 134
Struts Menu 128
Tag-Libraries 129
Validator 124
View 119
Internationalisierung 120
Tag-Libraries 120
Struts Menu 128, 164
Student 139
System
Objekt-orientiertes 16
Seiten-orientiertes 16
Struktogramm-orientiertes 16
Zeitachsen-orientiertes 16

T

Tag-Libraries 120, 129
Tele-
Learning 13
Teaching 13
Tutoring 13, 18
Templates 17
Thread 179
Tracking 51
Tracking Model 70
Trennzeichen 176
Tutor 139
Tutoring
begleitended 22
individuell 151
Lerntyp-bezogenes 22
persönliches 21, 22
programmiertes 22
Tutoring System 18
intelligentes 19

U

Uniform Resource Identifier 46
URL .... *siehe* Uniform Resource Identifier

V

Validator ..... *siehe* Struts Validator
Vector Space Search Algorithmus 179

Versuch ..... *siehe* Attempt
Verzeichnis, temporäres 178
View 119

W

WBT ..... *siehe* Web Based Training
Web Based Training 11, 18
WebCT 94
WebCT Campus ..... *siehe* WebCT
WebCT Vista ..... *siehe* WebCT
Wiederverwendbarkeit 42
Wort-
    Stamm
        Reduktion 179

X

Xerces2 Java Parser 172
XML
    Elemente 49

Z

Zwischenspeicher 178